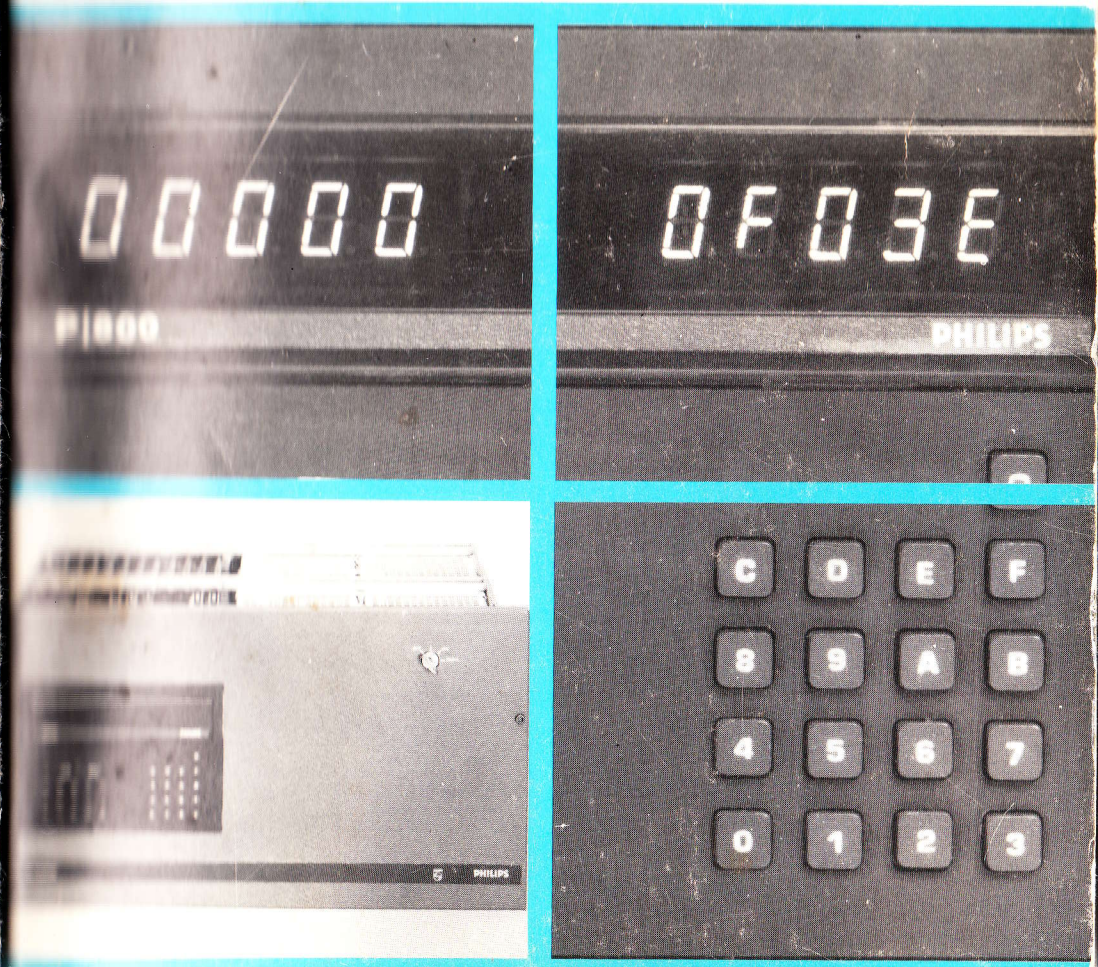
Data Systems

PHILIPS

P854M
System Handbook



P854M
System Handbook

A Publication of:
Philips Data Systems
Department SSS - T & D
P.O. Box 245, 7300 AE Apeldoorn, The Netherlands.

Publication Number 5122 991 31231

November 1981

Printed in the Netherlands.

Preface

This manual contains the description of the P854M mini computer.
It describes the central processor, memories, control units, interface signals and the internal organization of the system.
Moreover, a survey is given of standard peripherals and standard system software available to allow the customer to configurate his system according to the application.

Great care has been taken to ensure that the information contained in this manual is accurate and complete. Should a user, however, find any errors or omissions, or wish to suggest improvements, he is invited to write his comments on the sheet provided at the end of this book and send it to :

Philips Data Systems
Department SSS - T & D
P.O. Box 245, 7300 AE Apeldoorn, The Netherlands.

Table of Contents

1 Introduction

The P854M is one of the newest additions to the P800 range of 16-bit minicomputers on Eurocards.
Latest developments in LSI technology combined with proven software make the P854M a compact, but powerful and extremely versatile machine, suitable for a wide range of applications.

The major control and processing functions are contained on three double Eurocards. Two cards contain the microprogram-controlled central processing unit (CPU), a V24 serial contro unit and the control panel interface;a third card contains the memory management unit (MMU), a double I/O Processor with 16 sub-channels, and the logic for the ´stop on preset address´ function.

MEMORY

The P854M can address up to 512k words of memory. It uses MOS master-slave memory, i.e. one memory control card controls up to 8 memory slave cards, each containing 64k words of memory with a write access time of 173 nsec, a read access time of 445 nsec, and a read/write cycle time of 488 nsec.

MEMORY MANAGEMENT UNIT

The Memory Management Unit (MMU) provides the P854M with a virtual memoryaddressing system, based on the paging principle. Application programs aresplit up in up to sixteen 2k-word blocks, called pages. A 16-bit program address is translated dynamically by the MMU into a 20-bit physical address. In addition, the MMU provides memory protection per page.

UPL BUS

All interface signals are transferred by the asynchronous UPL (Unified Product Line) Bus. The bus is implemented on the back panel of the mounting box, which offers the advantage that extra memory modules or control unit cards can easily be plugged into an existing system.
The UPL bus is an extension of the P851M General Purpose Bus ´S´.

CONTROL PANELS

Two types of control panel are available: a Hand Held Control Panel, with a 4-digit hexadecimal display, and an Extended Control Panel with two 6-digit hexadecimal displays.
Via the control panel buttons and switches all memory locations and 16 general purpose register can be addressed, read and changed; programs can be started, executed step by step or interrupted, and a ROM bootstrap can be loaded into memory. A ´stop on preset address´ facility is available to stop the CPU on access to a specified memory address.

MOUNTING BOXES

Mounting boxes for the P854M fit in standard 19" racks. Two types of mounting boxes are available: one with 12 card slots and one with 22 card slots. Power supply units are included in the mounting boxes.

CONTROL UNITS

A wide range of peripheral control units and devices is available for connection to the system. Control units may be connected to UPL Bus for character or word transfers via the Programmed Channel under CPU control, or via an I/O Processor which controls block transfers between control unit and memory. Control units with direct memory access facility can be connected directly to the UPL Bus.
Up to 6 extra I/O Processors (with 8 sub-channels each) can be connected to the P854M.

Several data communication control units can be used with the P854M, providing facilities for asynchronous or synchronous operation, half duplex or full duplex transmission, point-to-point or multi-point connections and transmission speeds from 50 to 100,000 bps. Each data communication control unit is located on a single card which plugs directly into the bus.

For process control applications, an interface system for signal processing is available: the Modular Input Output System (MIOS), which can be configured to suit any analog or digital application, without additional equipment.

INSTRUCTION SET

The P854M executes the standard P800 instruction set, extended with instructions for bit string handling, character string handling and address handling. This powerful and comprehensive instruction set includes the following groups of instrucions:

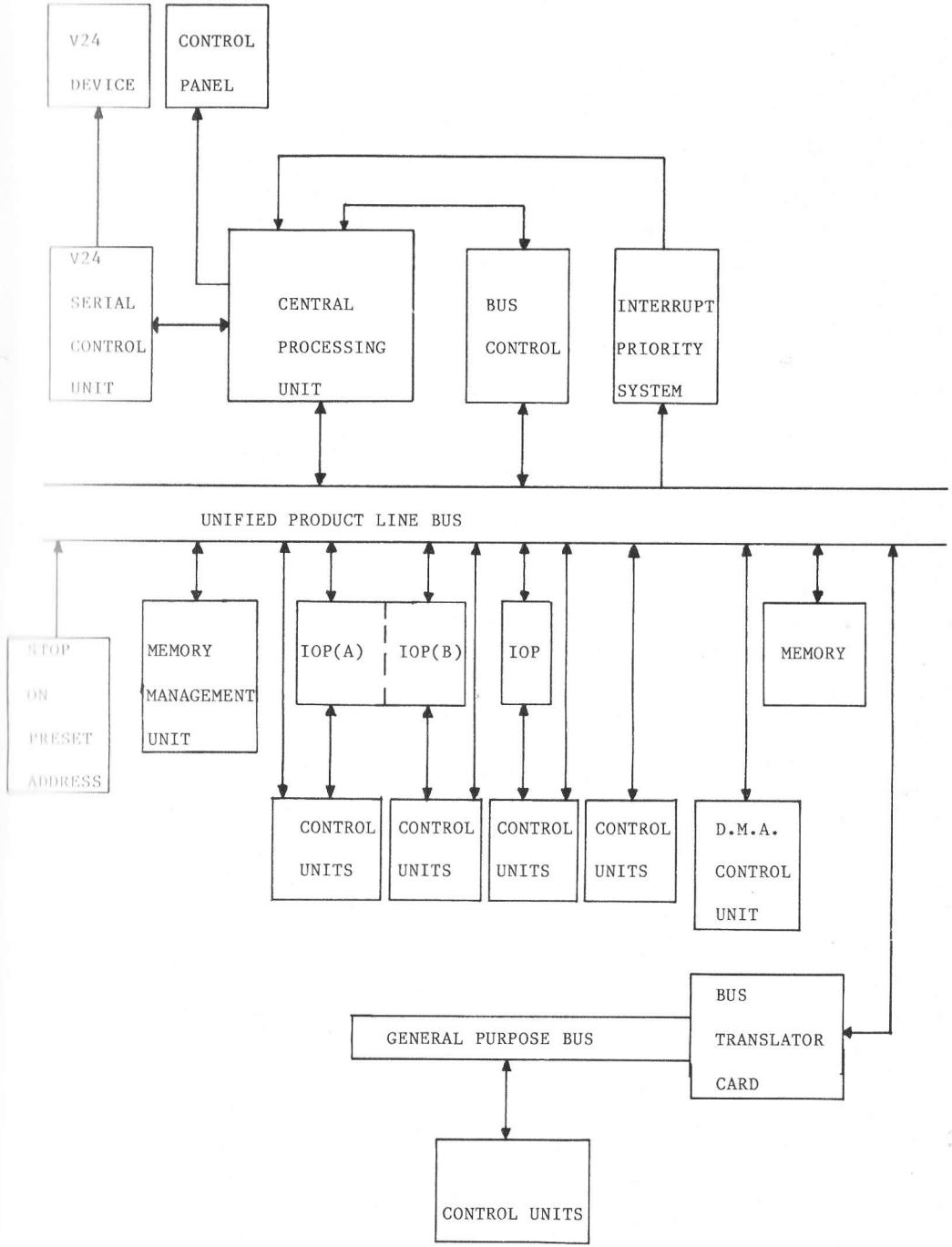
- load/store instructions
- arithmetic instructions
- logical instructions
- shift instructions
- branch instructions
- table handling instructions
- extended transfer instructions
- control instructions
- input/output instructions
- character handling instructions
- bit string handling instructions
- character string handling instructions
- address handling instructions

SYSTEM SOFTWARE

The main software operating system for the P854M is the Multi Application System (MAS). This disc oriented system supports real-time processing and batch processing functions, with dynamic allocation of all system resources. Functionally, MAS is divided into three support areas: operating support, data support and programming support. Operating support is provided by the Multi Application Monitor (MAM) which controls program execution and system operation; data support is provided by the Transaction-oriented Disc File Management (TDFM) facility and the data communication package TMS (Transmission Management System). Programming support incorporates compilers and translators (Assembler, FORTRAN, BASIC, RTL/2) and other programming aids (update package, linkage editor, macro processor, debugging package, librarian processor).

Compatibility throughout the P800 range makes it possible to run existing system software and application software on the P854M.

The figure on the opposite page shows the main system components and units thatcan be connected to the system.



2 Central Processing Unit

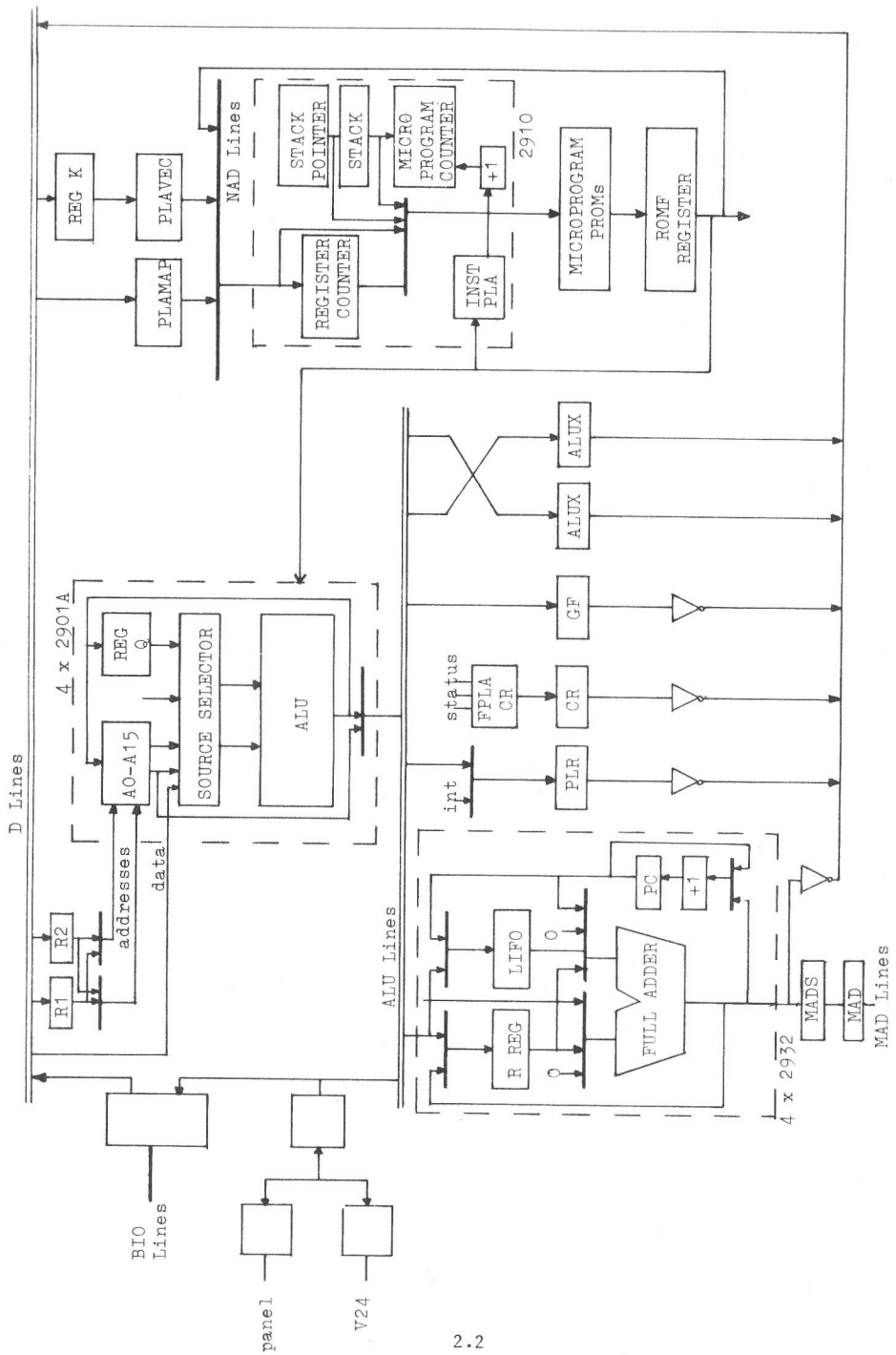
The P854M central processing unit is a LSI processor, contained on two double Eurocards, together with the V24 serial control unit. A third double Eurocard contains the Memory Management Unit (MMU) and a double I/O processor.

The CPU is controlled by microprogram instructions, contained in 7 PROM ICs and executed by a four-chip LSI processing unit, one chip with control logic for microprogram addressing and a four-chip memory addressing unit.
The CPU card is connected directly to the UPL bus.

The P854M executes the standard P800M instruction set, extended with instructions for bit string handling, character string handling and address loading.

Features of these CPU´s are:

- 16 hardware registers, 14 of which are fully programmable
- UPL bus controller- extended P800 standard instruction set
- 61 interrupt levels
- general purpose Initial Program Loader
- double I/O processor with 16 sub-channels
- DMA facility
- memory management allowing addressing beyond 32k
- addressing facility up to 512k words
- direct access (read/write) for up to 224 external registers
- interface for Floating Point Processor
- V24 serial control unit, speed selectable
- automatic diagnostic facility
- power failure / automatic restart
- real time clock, line frequency
- control panel interface



OPERATION

The figure on the opposite page shows the main components of the CPU. Their function and operation are:

Scratch Pad and Arithmetic Unit

The Scratchpad and Arithmetic Unit logic is contained in 4 LSI circuits type 2901A. The four circuits are connected in parallel for 16-bit operation. The main logic elements of the 2901A circuits are:

- Scratch Pad
- Source Selector
- Q Register
- Arithmetic Logic Unit

Scratch Pad

The Scratch Pad consists of sixteen 16-bit registers, comprising 15 working registers (A0 to A14) and the stack pointer (A15). Registers A1 to A15 are program addressable.Registers A1 to A14 are addressed from the program instruction format and may be used to hold one or both of the operands of an instruction, and possibly the result; they may also be used as addressing and indexing registers.

Register A15 is used as a stack pointer by the interrupt system; it may also be addressed from the instruction format in the same manner as registers A1 to A14 (in system mode only).

The scratch pad registers receive input from the BIO lines of the General Purpose Bus via the internal D bus; registers are addressed from the D bus via two 4-bit registers R1 and R2.

Source Selector

The Source Selector is used as multiplexor for the inputs to the Arithmetic Logic Unit. Controlled by microprogram instructions, it selects one out of eight possible input combinations from five operand sources.

Q Register

The Q register is used in multiplication and division operations and as accumulator. Its three modes (shift up, shift down, no shift) are controlled by microprogram instructions.

Arithmetic Logic Unit

The Arithmetic Logic Unit (ALU) performs arithmetic and logic operations on two 16-bit operands; the result is a single 16-bit word, sent to the ALU bus. The eight ALU functions (three arithmetic functions and five logic functions) are controlled by microprogram instructions.

Memory Addressing Logic

The Memory Addressing Logic consists of four LSI circuits type 2932, connected in parallel for 16-bit operation. The main logic elements are:

- LIFO Stack
- R Register
- Program Counter
- Full Adder

LIFO Stack

The LIFO Stack is a 17-word last-in/first-out stack, consisting of an input multiplexor, a 17-word random access memory, and a stack pointer. The stack is used for storing subroutine return addresses, and as a multi-purpose auxiliary memory.

R Register

The R register is an auxiliary register, used for holding either the ALU lines contents or the Full Adder output.

Program Counter

The Program Counter is a 15-bit register, which holds the current memory address. At the end of every instruction its contents are incremented by 1 (i.e. the actual address value is incremented by 2 to point to the next word).

Full Adder

The Full Adder and the two multiplexors at its input are controlled by microprogram instructions. The Full Adder output is either a memory address (sent to the MAD lines of the General Purpose Bus) or data for the internal D bus.

Microprogram Control Logic

The system microprogram controls overall CPU operation. The main logic elements of the microprogram control logic are:

- K Register
- PLAVEC Programmable Logic Array
- PLAMAP Programmable Logic Array
- Microprogram Sequensor
- Microprogram PROMs
- ROMF Register

K Register

The K register is used to hold either a complete instruction, or the most significant word of a double length instruction. The contents of that instruction are used to produce an address for access to the associated microprogram.

PLAMAP

PLAMAP is a programmable logic array, which senses the current CPU state via its inputs. In response to certain conditions, the preprogrammed address of a microprogram routine is sent to the NAD lines. It is used to define addressing routines or direct execution for short constant and register-to-register instructions.

PLAVEC

PLAVEC is a programmable logic array, which decodes an instruction from K register. The preprogrammed output is the address of a microprogram routine, which is sent to the NAD (Next Address) lines.

Microprogram Sequensor

The Microprogram Sequensor is a LSI circuit type 2910, which controls the addressing and execution of the microinstructions contained in the PROMs.
The Microprogram Sequensor consists of:

- a register counter, which receives the next address from the NAD lines
- a five word, 12-bit last-in/first-out stack, which stores the return addresses for subroutines and loops
- a microprogram counter, consisting of a 12-bit incrementor and a 12-bit register
- a programmable logic array, which decodes the 16 microinstructions controlling the Microprogram Sequensor
- a multiplexor, which selects the microprogram address source from its four inputs (NAD lines, register counter, LIFO stack or microprogram counter)

Microprogram PROMs

The microprogram consists of 2048 56-bit words, contained in 7 PROMs (Programmable Read Only Memories). It is addressed from the multiplexor in the microprogram sequensor; its output (i.e a microprogram instruction word) is sent to the ROMF register.

ROMF Register

The ROMF register is a 56-bit register, which holds the current microprogram instruction word. From this register CPU operation is controlled; the register contents are also sent to the NAD lines as possible source for the next address.

Program Status Word

The Program Status Word, holding the current CPU state, is contained in five register, which together form one word.
The five registers are:

- Program Level Register (PL)	6 bits
- condition Register (CR)	2 bits
- Enable Indicator	1 bit
- Extended System Mode Indicator	1 bit
- User Mode Indicator	1 bit

PL Register

The PL register consists of six bits, which hold the priority level of the current program. Input for this register comes either from the ALU output { current priority level) or from the interrupt system (new priority level after an interrupt).

CR Register

The CR register holds in two bits the result of, or the response to certain instructions. A 16-input programmable logic array provides preprogrammed codes as input for the CR register.

Enable / Extended System Mode / User Mode Indicators

The three one-bit indicators for Enable Interrupts, Extended System Mode and User Mode are set to indicate the corresponding CPU state. The indicators derive their input form the ALU lines and from direct inputs.

Internal Data Buses

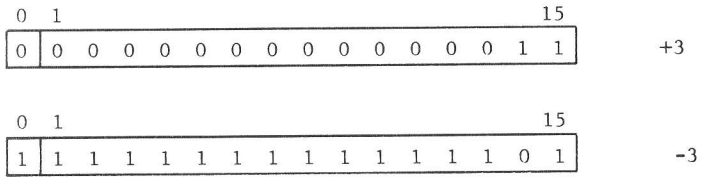
The CPU functional units are interconnected by two internal data buses.
The ALU bus receives the ALU output and provides input for the memory addressing logic and the machine state indicators; the ALU output can also be sent to the BIO lines of the UPL bus.
The D bus receives data from the BIO lines, and provides input for the scratch pad registers, the ALU, and the microprogram addressing logic.

ALUX Registers

The two 8-bit ALUX registers are used for direct transfers from the ALU bus to the D bus with exchanged byte contents (i.e. $ALU_{0-7} \rightarrow D_{8-15}$ or $ALU_{8-15} \rightarrow D_{0-7}$).
This direct transfer facility reduces the length of the data path in Load Character and Exchange Character instructions.

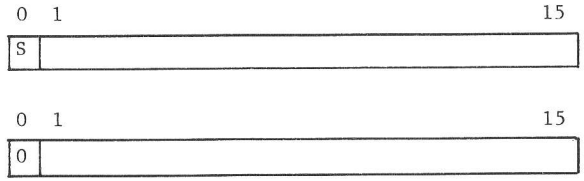
3 Basic Word

The basic data format within the system is a single precision integer contained in a 16-bit word. The word´s bit positions are numbered from 0 (most significant bit) to 15 (least significant bit).Bit 0 is used as a sign bit, bits 1 to 15 contain a binary numeric value. For positive values bit 0 is 0; negative values are represented by a ´1´ in bit 0,
and the 2´s complement of the corresponding positive value in bits 1 to 15.
The figure below shows the binary representation of +3 and -3:



For programming purposes the data word may be divided into two 8-bit characters which may be used independantly by certain instructions. Bits 0 to 7 of the word represent the left hand character and bits 8 to 15 the right hand character.

Double precision integer format is used for extended precision arithmetic operations. Double precision data are represented by two consecutive words. The most significant bit of the second word has to be 0, so 31 bits (including the sign bit) are available for data representation, as shown below:



4 Memory

The standard memory type for the P854M is the master-slave memory, consisting of a memory control card, and one or more slave cards.

The slave cards contain 64k 21-bit words of semi-conductor memory. The 21-bit word consists of 16 data bits and 5 bits for error detection and correction of single bit errors.

The master-slave memory has a typical write access time of 173 nsec, a read access time of 445 nsec, and a read/write cycle time of 488 nsec.

The memory control card can control up to 8 slave cards, which gives a maximum memory capacity of 512k words.

A separate battery back-up is recommended, to prevent loss of data when a power failure occurs.

WORD/CHARACTER MODE

The master-slave memory operates in either word or character mode. In word mode the contents of the input/output lines are set from or written into the corresponding bits of the addressed memory word; in character mode the contents of the least significant eight input/output lines are set from or written into either the left hand or right hand character of the addressed memory location. In character mode the unused character is left unaltered.
Figure 4.1 shows character mode operation for a left hand and a right hand character.

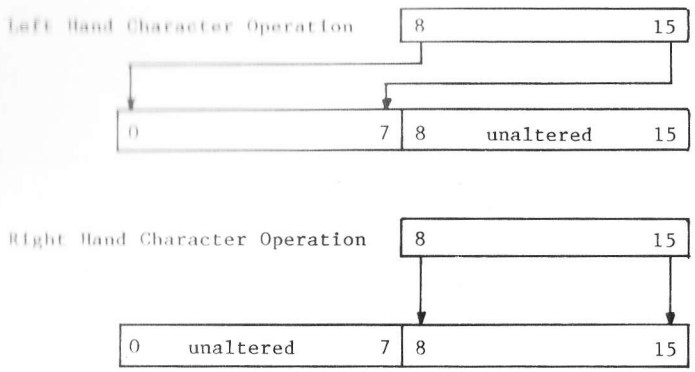


Figure 4.1 Operation of memory in character mode

MEMORY ADDRESSING

Depending upon the type of instruction or operation being carried out the memory may be addressed in words or characters for programming purposes.
Bits 0 through 14 of the address are used to access memory in word mode. Bit 15 is in that case not used.
When operating in character mode, all sixteen address bits are used. Bits 0 through 14 address the memory location and bit 15 addresses the character in the word. When bit 15 is 0 the left hand character is addressed, when bit 15 is 1 the right hand character is addressed.

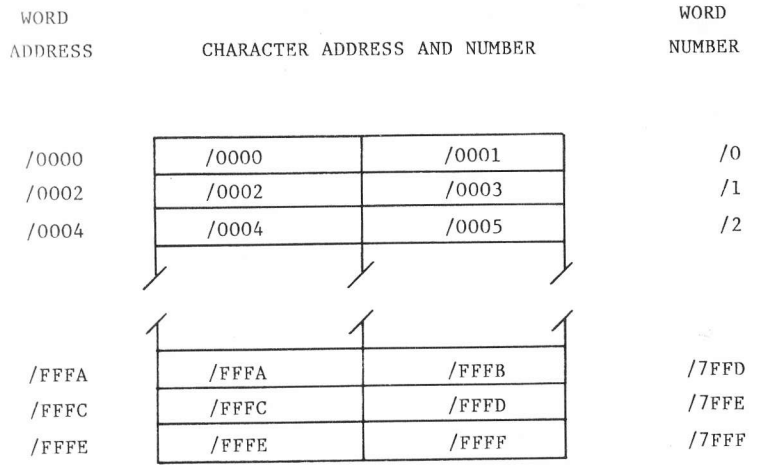


Figure 4.2 Layout of words and character in memory

Memory addressing over 32k is entirely transparent to the user; the logicaladdresses (<32k) in a user program are translated into physical addresses>32k by the Memory Management Unit (see chapter 5).

5 Memory Management Unit MMU

The Memory Management Unit (MMU) allows memory addressing extension over 32k words, up to 512k words. It permits dynamic program relocation in multitask programming under control of the Multi Application Monitor, Disc Real Time Monitor or Basic Real Time Monitor and offers memory protection and error detection facilities.

For memory addressing through the MMU, the system considers the memory as consisting of n blocks of 2k words, called pages. When a system or user program is called, it is loaded into memory, split up in pages of 2k, which do not need to be contiguous. Moreover, only those parts of the program are loaded which are required for execution. A system or user program must not exceed 32k words (16 pages).

ADDRESS TRANSLATION

The MMU translates the 16-bit relative addresses in a user program into 20-bit physical addresses by means of a page table of 16x16 bits, which is built for each program. The 16-bit logical address is divided in two parts: a logical page address taken from bits 0-3, and a displacement value taken from bits 4-15. The logical page address points to a page table word, containing a 12-bitphysical page address in bits 0-5 (least significant part) and bits 10-15 (most significant part). The 8-bit physical page address is combined with the 12-bit displacement value to form a 20-bit physical address.
The address translation for P854M is shown schematically in Figure 5.1.

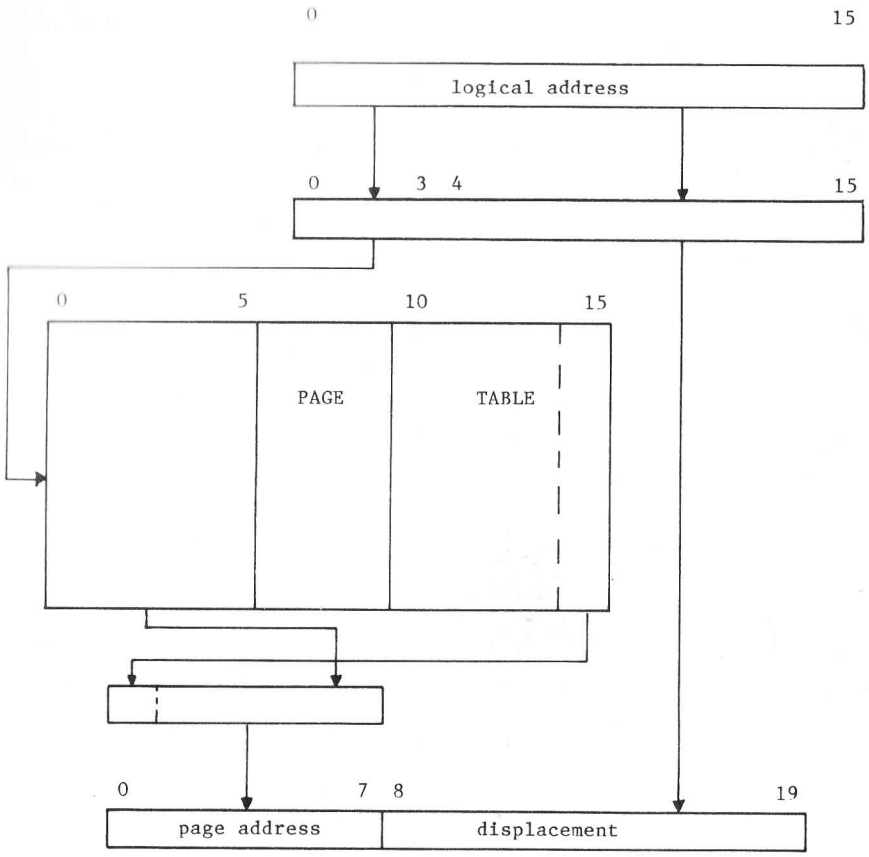
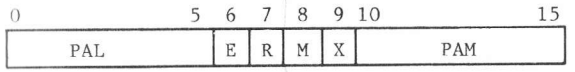


Figure 5.1 MMU Address Translation

In system mode (see chapter 12), the MMU does not translate addresses, as the system software uses absolute addressing. To transfer in this mode data from a system area to a user area, special instructions are used (Move Table fromSystem to User, and Move Table form User to System).
In extended system mode (see chapter 12), the MMU translates all addresses; for programs running in the first 32k words of memory, the MMU performs a dummytranslation.

Page Table

The page table consists of sixteen 16-bit words, containing the page description of a program. The monitor loads and stores the segment table by means of the instructions Page Table Load (TL) and Page Table Store (TS).
The format of a page table word is:



bit 0-5 PAL Physical page address (least significant part).

bit 6 E Page error bit. This bit is 0 for memory resident pages of the running user program only. An attempt to access a page with E=1 causes a page fault trap (see below).

bit 7 R Read only page indicator. When R=1, the page is protected against overwriting. An attempt to write in that page causes a page fault trap (see below).
A read only page can be shared by several user programs.

bit 8 M Modified page indicator. This bit, initially reset to 0, is set to 1 after write access (i.e. after changing the contents of the page). If M=1 the page must be restored on disc before a new page is loaded; if M remains 0, the page can be overwritten.

bit 9 X reserved

bit 10-15 PAM Physical page address (most significant part).

Page Fault Handling

A page fault trap is generated when an attempt is made to access a missing or wrong page in user mode, or to write into a protected page.
If a page fault trap occurs, a trap action is activated. This mechanism is described in detail in chapter 10. For the page fault trap, in addition to the P and PSW registers (program counter and program status word), a third word is stored in the system stack. This word contains in bits 4-7 the logical page number, all other bits are 0.

6 Instruction Set

The instruction set gives the programmer the ability to carry out all the functions necessary to program the system efficiently.

There are 13 basic instruction groups:

- Load/Store Instructions
- Arithmetic Instructions
- Logical Instructions
- Character Instructions
- Branch Instructions
- Shift Instructions
- Control Instructions
- Input/Output Instructions
- External Transfer Instructions
- Move Table Instructions
- Bit String Handling Instructions
- Character String Handling Instructions

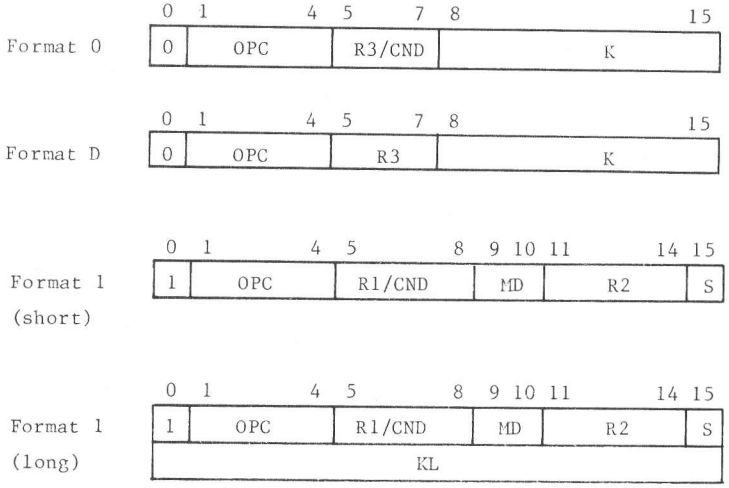
Within these groups efficiency is ensured by the possible use of up to eight different methods of forming one of the instruction's operands, the method to be used being chosen by the programmer with reference to the memory and timing requirements of any particular program.

Two formats for instruction layouts are used and where necessary two words are used to define an instruction.

INSTRUCTION FORMATS

Two general instruction formats are possible, defined within the instruction by the most significant bit, bit 0, of the instruction word. Where instructions consist of two words the format bit is the most significant bit of the first word.

Format 0 instructions are always short, that is one word. Format 1 instructions may be short or long, one or two words.
An extension of format 0 is the D format, differing from the normal format 0 only in the way the operands are specified.
The layout of the instruction formats is shown below:



The function of the instruction fields is:

OPC 4 bits, the pattern of which defines the instruction to be carried out

R1 4 bits, specifies the working register to be used by the instruction (A0 - A15). It may contain one of the operands to be used and may also be used to hold the result of the instruction. In certain cases with R1=0, the addressed register (the P register) will not be used; in these cases R1=0 qualifies the operation code OPC, to define a different instruction than when R1≠0.
The actual register number is specified in the instruction format with the most significant bit in bit 8 and the three least significant bits in bits 5-7.

R2 4 bits, specifies the second working register to be used (A1-A15). It may contain the second operand, or hold an address to be used in

forming the operand. If R2=0, no second working register is specified; it indicates another method of forming the second operand. The actual register number is specified in the instruction format with the most significant bit in bit 14 and the three least significant bits in bits 11-13.

R3 3 bits, specifies the working register to be used (A0 - A7). It may contain one of the operands and may also be used to hold the result of the instruction. In certain cases with R3=0, the addressed register (the P register) will not be used; in these cases R3=0 qualifies the operation code OPC, to define a different instruction than when R3≠0.

CND 3 bits, specifies the condition which must exist for a particular instruction to be carried out. Used to qualify conditional branch instructions. In format 1 instructions, CND is specified in bits 5-7.

MD 2 bits, specifies the addressing mode to be used for forming the second operand of an instruction where this is applicable.

S 1 bit, applicable to certain instructions using memory. When S=1 it specifies that the result of the instruction is to be stored in the memory address specified in the instruction. When S=0, the result is placed into the working register specified by R1.

I 8 bits, these bits are used to specify the operand in format 0 instructions, and include short constant operands and short displacements for relative branch instructions. This field is also used to specify counts for shift instructions, or device addresses for I/O instructions; in these cases a part of the field may be used to qualify the operation code.

KL 16 bits in the second word of a double length instruction, specifies a long constant or an address.

F 8 bits in format D instructions, qualifying the OPC.

FORMING THE OPERAND

Many of the instructions may use various methods of forming one of the operands to be used. In all, nine methods of forming an operand are available, governed by the values of the format, mode and R2 field of the instruction layout.

Figure 6.1 lists the nine methods of forming an operand; a brief description of each method is given following the figure.

Type	Format	Mode	R2	
T1	1	00	-	Reg./Reg.
T2	1	01	R2=0	Long Constant
T3	1	01	R2≠0	Address in Reg R2
T4	1	10	R2=0	Address in next word
T5	1	10	R2≠0	Indexed
T6	1	11	R2=0	Indirect
T7	1	11	R2≠0	Indexed Indirect
T8	0	-	-	Short Constant
T8	D	-	-	Addresses in registers

Figure 6.1 Operand Definition

Depending on the instruction type, the operand defined by one of the nine method described above may be the first or second of two operands, or the only operand. The nine methods of defining an operand are:

T1 Register/Register Format 1 (short)
The register specified by R2 contains an operand.

T2 Long Constant Format 1 (long)
The second instruction word (all 16 bits) contains an operand.

T3 Address in Register Format 1 (short)
An operand is held in memory. The memory address of the operand is the value in the register specified by R2

T4 Address in Next Word Format 1 (long)
An operand is held in memory. The memory address of the operand is the value in the second instruction word

T5 Indexed Address in Next Word Format 1 (long)
An operand is held in memory. The meory address of the operand is found by adding the value in the register specified by R2 to the value in the second instruction word

T6 Indirect Address in Next Word Format 1 (long)
An operand is held in memory. The memory address of the operand is also held in memory, at the address specified in the second instruction word

T7 Indexed Indirect Address in Next Word Format 1 (long)
An operand is held in memory. The memory address of the operand is also held in memory, at the address found by adding the value in the register specified by R2 to the value in the second instruction word

T8 Short Constant Format 0
An operand is defined in the least significant 8 bits of the instruction

T8 Parameters in Registers Format D
The parameters for these instructions (addresses, string lengths, character constants) are specified in scratch pad registers (A1 - A4)

INSTRUCTION TIMING

The timing of the instructions depends on various factors: the type of instruction itself, the memory, the method of forming the operand and the number of memory cycles required.
The instruction set offers the possibility of very rapid execution times where single word register/register or short constant operations are employed whilst the more complex register/memory instructions save execution time when compared with the routines they may replace.
Execution time is also reduced in the case of conditional instructions by carrying out the conditional check immediately after accessing the instruction and then only continuing if the required conditions are satisfied.

TRAP ACTION

The use of an invalid instruction causes the activation of the Trap action which consists of the following basic actions:

- the CPU does not attempt to carry out the instruction
- information about the instruction address and processor status is saved in the stack
- interrupts are inhibited
- a user mode flag is reset when working in user mode
- an indirect branch is made to address /7E for a trap routine

THE INSTRUCTION SET

The instructions, together with their mnemonic and addressing type(s) are listed below.

Load/Store Instructions

LD	Load	T4,T5
LD*	Load	T6,T7
LDR	Load Register	T1
LDR*	Load Register	T3
LDKL	Load Constant	T2
LDK	Load Constant	T8
ST	Store	T4,T5
ST*	Store	T6,T7
STR	Store Register	T3
ML	Multiple Load	T4,T5
ML*	Multiple Load	T6,T7
MLR	Multiple Load	T3
MLK	Multiple LoadConstant	T2
MS	Multiple Store	T4,T5
MS*	Multiple Store	T6,T7
MSR	Multiple Store Register	T3

LDA	Load Address	T1
EL	Extended Load	T4,T5
EL*	Extended Load	T6,T7
ELR	Extended Load Register	T3
ES	Extended Store	T4,T5
ES*	Extended Store	T6,T7
ESR	Extended Store Register	T3

Arithmetic Instructions

AD	Add	T4,T5
AD*	Add	T6,T7
ADS	Add	T4,T5
ADS*	Add	T6,T7
ADR	Add Register	T1
ADR*	Add Register	T3
ADRS	Add Register	T3
ADK	Add Constant	T8
ADKL	Add Constant	T2
SU	Subtract	T4,T5
SU*	Subtract	T6,T7
SUS	Subtract	T4,T5
SUS*	Subtract	T6,T7
SUR	Subtract Register	T1
SUR*	Subtract Register	T3
SURS	Subtract Register	T3
SUK	Subtract Constant	T8
SUKL	Subtract Constant	T2
MU	Multiply	T4,T5
MU*	Multiply	T6,T7
MUR	Multiply Register	T1
MUR*	Multiply Register	T3
MUK	Multiply Constant	T2

DV	Divide	T4,T5
DV*	Divide	T6,T7
DVR	Divide Register	T1
DVR*	Divide Register	T3
DVK	Divide Constant	T2
DA	Double Add	T4,T5
DA*	Double Add	T6,T7
DAR	Double Add Register	T1
DAR*	Double Add Register	T3
DAK	Double Add Constant	T2
DS	Double Subtract	T4,T5
DS*	Double Subtract	T6,T7
DSR	Double Subtract Register	T1
DSR*	Double Subtract Register	T3
DSK	Double Subtract Constant	T2
C2	Two's Complement	T4,T5
C2*	Two's Complement	T6,T7
C2R	Two's Complement Register	T3
IM	Increment Memory	T4,T5
IM*	Increment Memory	T6,T7
IMR	Increment Memory Register	T3
NGR	Negate Register	T1
CM	Clear Memory	T4,T5
CM*	Clear Memory	T6,T7
CMR	Clear Memory Register	T3
CW	Compare Word	T4,T5
CW*	Compare Word	T6,T7
CWR	Compare Word Register	T1
CWR*	Compare Word Register	T3
CWK	Compare Word Constant	T2

Logical Instructions

AH	Logical AND	T4,T5
AH*	Logical AND	T6,T7
ANS	Logical AND	T4,T5
ANS*	Logical AND	T6,T7
ANR	Logical AND Register	T1
ANR*	Logical AND Register	T3
ANRS	Logical AND Register	T3
ANK	Logical AND Constant	T8
ANKL	Logical AND Constant	T2
OR	Logical OR	T4,T5
OR*	Logical OR	T6,T7
ORS	Logical OR	T4,T5
ORS*	Logical OR	T6,T7
ORR	Logical OR Register	T1
ORR*	Logical ORRegister	T3
ORRS	Logical ORRegister	T3
ORK	Logical OR Constant	T8
ORKL	Logical ORConstant	T2
XR	Exclusive OR	T4,T5
XR*	Exclusive OR	T6,T7
XRS	Exclusive OR	T4,T5
XRS*	Exclusive OR	T6,T7
XRR	Exclusive OR Register	T1
XRR*	Exclusive OR Register	T3
XRRS	Exclusive OR Register	T3
XRK	Exclusive OR Constant	T8
XRKL	Exclusive OR Constant	T2
TM	Test Mask	T1
TNM	Test Not Mask	T1

CI	One's Complement	T4,T5
CI*	One's Complement	T6,T7
CIS	One's Complement	T4,T5
CIS*	One's Complement	T6,T7
CIR	One's Complement Reg.	T1
CIR*	One's Complement Reg.	T3
CIRS	One's Complement Reg.	T3

Character Handling Instructions

LC	Load Character	T4,T5
LC*	Load Character	T6,T7
LCR	Load Character Register	T3
LCK	Load character Constant	T2
SC	Store Character	T4,T5
SC*	Store Character	T6,T7
SCR	Store Character Register	T3
CC	Compare Character	T4,T5
CC*	Compare Character	T6,T7
CCR	Compare Character Reg.	T3
CCK	Compare Character Const.	T2
ECR	Exchange Character Reg.	T1

Branch Instructions

AB	Absolute Branch	T8
ABL	Absolute Branch	T2
ABI	Absolute Branch Indirect	T4,T5
ABI*	Absolute Branch Indirect	T6,T7
ABR	Absolute Branch Register	T1
ABR*	Absolute Branch Register	T3
RB	Relative Backward Branch	T8
RF	Relative Forward Branch	T8

CF	Call Function	T2
CFI	Call Function Indirect	T4,T5
CFI*	Call Function Indirect	T6,T7
CFR	Call Function Register	T1
CFR*	Call Function Register	T3
RTN	Return (A1...A14)	T3
RTN	Return (A15)	T3
EX	Execute	T4,T5
EX*	Execute	T6,T7
EXR	Execute Register	T1
EXR*	Execute Register	T3
EXK	Execute Constant	T2

Shift Instructions

SLA	Left Arithmetic Shift	T8
SRA	Right Arithmetic Shift	T8
SLL	Left Logical Shift	T8
SRL	Right Logical Shift	T8
SLC	Left Circular Shift	T8
SRC	Right Circular Shift	T8
SLN	Left Shift/Normalize	T8
SRN	Right Shift/Normalize	T8
DLA	D.Left Arithmetic Shift	T8
DRA	D.Right Arithmetic Shift	T8
DLL	D.Left Logical Shift	T8
DRL	D.Right Logical Shift	T8
DLC	D.Left Circular Shift	T8
DRC	D.Right Circular Shift	T8
DLN	D.Left Shift/Normalize	T8
DRN	D.Right Shift/Normalize	T8

Control Instructions

ENB	Enable	T8
HLT	Halt	T8
RIT	Reset Internal Interrupt	T8
INH	Inhibit Interrupts	T8
LKM	Link to Monitor	T8
SMD	Set Mode	T8

Input/Output Instructions

CIO	Control Input/Output	T8
INR	Input to Register	T8
OTR	Output from Register	T8
SST	Send Status	T8
TST	Test Status	T8

External Transfer Instructions

WER	Write External Register	T8
RER	Read External Register	T8

TL	Segment Table Load	T4,T5
TL*	Segment Table Load	T6,T7
TLR	Segment Table Load Reg.	T3

TS	Segment Table Store	T4,T5
TS*	Segment Table Store	T6,T7
TSR	Segment Table Register	T3

Move Table Instructions

MVF	Move Table Forward	T8
MVB	Move Table Forward	T8
MVUS	Move Table User/System	T8
MVSU	Move Tabel System/User	T8

Bit String Handling Instructions

TSB	Test and Set Bit	T4,T5
TSB*	Test and Set Bit	T6,T7
TSBR	Test and Set Bit Reg.	T3
TRB	Test and Reset Bit	T4,T5
TRB*	Test and Reset Bit	T6,T7
TRBR	Test and Reset Bit Reg.	T3
TB	Test Bit	T4,T5
TB*	Test Bit	T6,T7
TBR	Test Bit Register	T3

Character String Handling Instructions

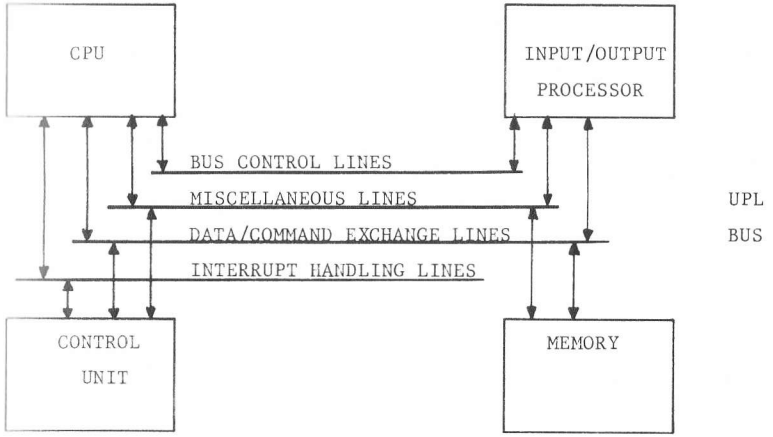
MVS	Move String	T8D
MVA	Move All	T8D
CS	Compare String	T8D
CA	Compare All	T8D

7 Unified Product Line Bus

The Unified Product Line (UPL) bus, consisting of data lines, addressing lines, interrupt lines and control lines, handles the exchanges made between the main units of the system. Four groups of signals provide four bus functions:

- bus control
- data/command exchanges
- interrupt handling
- miscellaneous functions

In order to gain the maximum efficiency from the bus, certain functions may occur concurrently. Bus control functions may occur during the current data or command exchange, interrupt handling is carried out entirely independant from other facilities once it has been initiated, and miscellaneous functions may occur at any time and without reference to any other bus function.



BUS CONTROL FUNCTIONS

Efficient use of the bus for data or command exchanges is organized by a bus controller within the CPU. This controller allocates bus cycles for data or command exchanges , on a priority basis, to units which are able to requestsuch cycles.

All units connected to the bus are designated as either master units or slave units.

Master units can request a bus cycle for data or command exchanges and control the subsequent exchange with another master or slave, or between two separate slaves.
Slave units cannot request a bus cycle; these units must wait until a master signals that an exchange is allowed.
A master unit may act as either master or slave, depending on the type of exchange.

Priority Chain

Bus cycle allocations are made on a priority basis, to avoid clash conditions when two or more master units request a bus cycle simultaneously.
A priority chain is hardware wired at installation time; within a standard system the CPU has the lowest priority, other master (such as I/O Processors) are given priority according to the system's requirements.
An enable signal from the CPU can overrule the priority order: masters to which the enable signal is connected, can only request a bus cycle when the enable signal is active, i.e. when the CPU is not using the bus.
The combination of the priority chain and enable signals provides three main priority levels:

- masters which can always issue a bus request (i.e. masters not connected to the priority signal)
- the CPU
- masters which can only issue a bus request when enabled by the CPU

The CPU automatically gets the bus whenever it is free, but the bus controller continues to monitor the bus request signal lines.
A bus request / master selection sequence consists of the following actions:

- a bus cycle request is issued by one or more master units
- the bus controller raises a scan signal, which is chained through all masters in order of priority; onward transmission is inhibited by the first master that requested the bus (if the enable signal from the CPU is connected to a master, that master can only inhibit the scan signal when the enable signal is active)
- the selected master unit signals its selection via the bus to all other masters
- any other lower priority masters requesting bus cycles remove their requests

- the selected master unit waits until the exchange paths within the bus are free
- when the bus is free, the selected master takes the bus and processes the exchange
- the selected master removes its 'master selected' signal from the bus, which allows a new bus request / master selection sequence to be carried out whilst the exchange takes place
  (a selected master can delay the removal of 'master selected' until it knows whether it wants the bus for the next exchange, but there must be enough time left for a bus selection sequence within the time necessary for the exchange)

Figure 7.1 shows schematically the bus priority and selection system, with the following signal names:

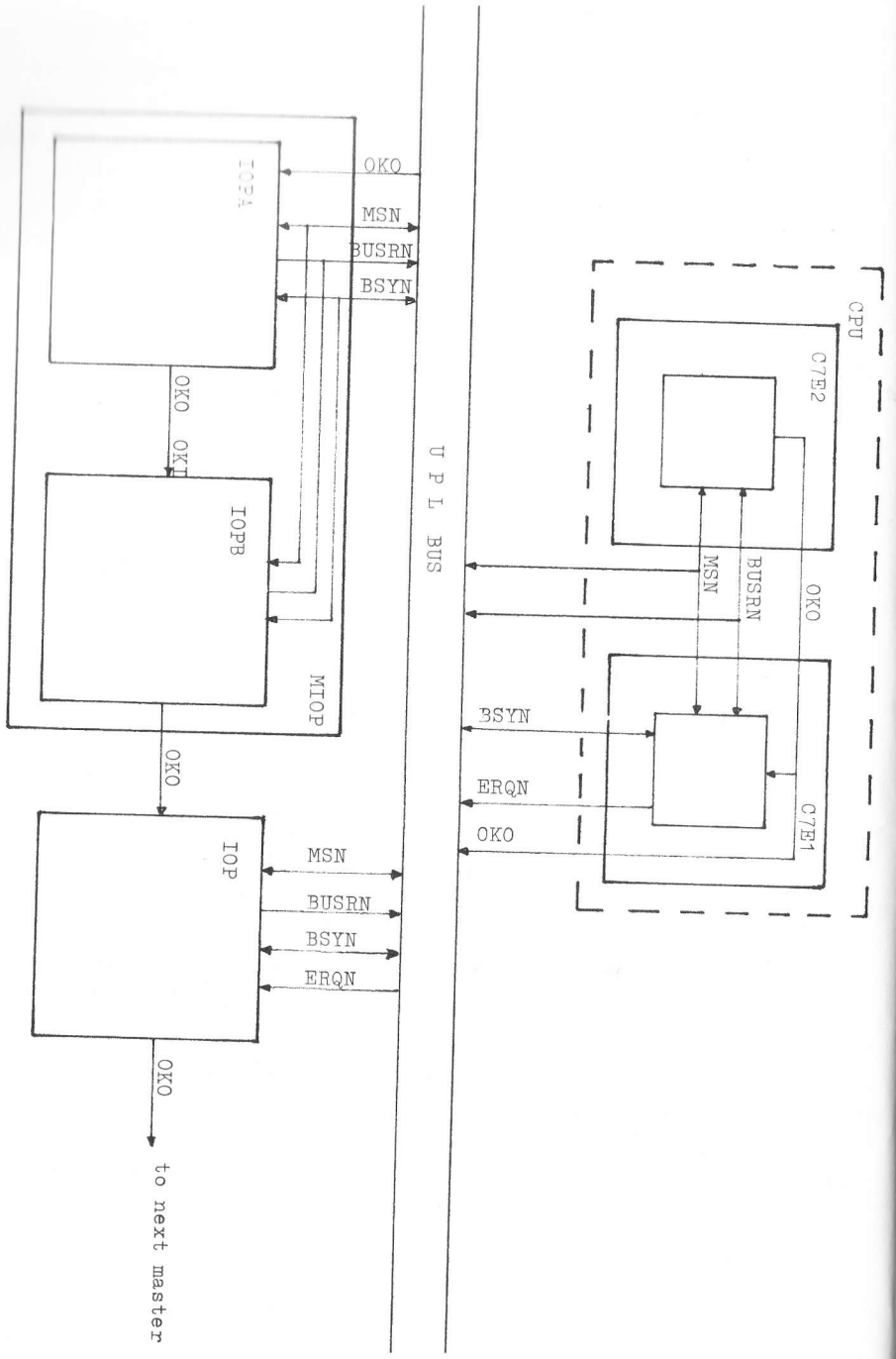
DEO/OKI - scan signal
BUSRN - bus request
BSYN - bus busy
MSN - master selected
ERQN - enable request

DATA OR COMMAND EXCHANGES

Data or command exchanges comprise data transfers with memory, command and response transfers with control units, and transfers with external registers.
Basically two types of exchanges are possible:

- exchanges between the controlling master and another unit, where the other unit is either a slave unit or a master acting as a slave, e.g.CPU to control unit or CPU to I/O Processor
- exchanges between two slave units under control of a master, e.g. an I/O Processor controlling an exchange between a control unit and memory

The timing of data or command exchanges is controlled by timing signals raised by the master unit carrying out or controlling the exchange. These timing signals validate the data on the bus address lines or bus data lines.
The bus includes a time out facility, which unblocks the priority system if there is no reply to a timing signal from a master within 10 usec.



INTERRUPT HANDLING

Interrupt handling is carried out independantly of other bus functions via separate bus interrupt lines. The CPU initiates a scan of the interrupt lines at the beginning of every instruction, and after each instruction step of some instructions. The overall operation of the interrupt system is covered in the chapter 'Interrupt System'.

BUS SIGNAL LINES

The signals and lines associated with the bus functions are listed below. N as last character of a signal name indicates an active low signal.

Bus Control Signals

BUSRN - Bus Request

This signal is raised by a master whenever it requires a bus data or command exchange cycle and bus requests are allowed.

ERQN - Enable Request

This signal is activated by the bus controller when the bus is not in use, no bus request has been issued, and no master has been selected. It enables the master to which it is connected to issue a bus request.

OKO / OKI - Check Requests

This signal is generated as OKO by the bus controller and received by the highest priority master as OKI. It is chained through all the masters in order of priority as OKO / OKI. Onward transmission of the signal is inhibited by the first master which receives the signal and is requesting a bus cycle, this master is then selected as the next master.

MSN - Master Selected

This signal is activated by a master which has been selected as the next master to indicate the selection to all masters. It is removed by the master concerned during its exchange cycle.

BSYN - Bus Busy

This signal is activated by the master which has been selected and is now carrying out an exchange cycle. It is removed on completion of the cycle to allow the next selected master to commence its exchange.

Data or Command Exchange Signals

Timing Signals

TMRN - Timing Master to Memory

This signal is sent by a master and is used to validate the data and address, and to control the timing of an exchange with memory.

TMPN - Timing Master to Peripheral

This signal is sent by a master and is used to validate the control unit address and data, and to control the timing of the exchange.

TMEN - Timing Master to External Register

This signal is sent by a master and is used to validate the data and address, and to control the timing of an exchange between a master and a unit containing an external register.

TSMN - Timing Slave to Master

This signal is sent from the IOP to the CPU during IOP initialization, or from a control unit or memory to an IOP during a bus exchange.

Bus Address Lines

MAD00 to MAD15, MADE0 to MADE7 - Address Lines

These lines carry the memory address, register address, or peripheral address and the required function, during any exchange and are qualified by the timing signals from a master:

1. Memory Exchanges (qualified by TMRN)

MAD00 to MAD14 - These lines carry the 15-bit memory address required to access up to 32k words of memory.

MAD15 - This line is only significant in character operations and is used to define the character within the addressed word which is to be used.

MADE0 to MADE7 - These lines enable the extension of memory addressing to 512k words (MADE0-MADE3 are always 0)

2. External Register Exchanges (qualified by TMEN)

MAD08 to MAD15 - These lines carry the 8-bit register address required to access up to 224 external registers.

MAD04 - This line is used to indicate a read or write operation the addressed register.

3. Peripheral Control Unit Exchanges (qualified by TMPN)

MAD10 to MAD15 - These lines carry the 6-bit device address required to access up to 64 control units.

MAD04 - This line is a function line, used to indicate the direction of the exchange.

MAD08 - This line is a function line, used to indicate whether t exchange is a data exchange or a command or status excha

MAD09 - This line is reserved for special functions.

MAD03 - This line is used to indicate whether the current word c character exchange is the last one of a block of exchan

Bus Data Lines

BI000N to BI015N - Input/Output Lines

These lines are the 16 input/output lines used to carry data between units making an exchange.

ACK - Accept
This signal is sent from a control unit to indicate that it accepts the request to carry out a designated function.

WRITE - Write
This signal is raised by a master controlling an exchange with memory to indicate that the exchange is a write to memory. When the signal is not present a read from memory cycle is indicated.

CHA - Character
This signal is sent from a master to memory to indicate that the requested exchange is to be carried out in character mode.

Bus Interrupt Lines

BCI - Bus Coded Interrupt
This signal line transmits serial interrupts from UPL control units

INCL - Interrupt Clock
This signal clocks the serially transmitted interrupt codes on the BCI line

Miscellaneous Signals

CLEARN - Clear
This sigal is sent from the CPU to all units connected to the bus and initiates a general reset of all such units.

RSLN - Reset Line
This signal is sent during the power on or power restoration sequence and is used within the system to ensure an orderly commencement or resumption of operation without loss of data.

PWFN - Power Failure
This signal is sent during the power off or power failure sequence and is used within the system to ensure an orderly run down of operation without loss of data.

8 Interrupt System

The P854M has been implemented with a hardware interrupt decoding system to allow for the switching from one program part to another upon reception of an interrupt signal fronm the CPU or a control unit.
Efficient handling of these signals is ensured by the hardware in conjunction with the system software, taking care that interrupt signals are serviced in the correct order of priority with complete recovery facilities to the interrupted program once the interrupt has been serviced.

HARDWARE INTERRUPT HANDLING

The interrupt system accepts the connection of 61 interrupt levels numbered from 0 through 60, where level 0 has the highest priority and level 60 the lowest.
Four interrupt levels are reserved for internal interrupts which are prewiredfrom CPU logic and have the highest priority. The remaining 57 levels are usedfor external interrupts, i.e. from control units.

The CPU has a Priority Level Register PLR which contains the interrupt level of the program interrupt being serviced. This level may range from 0 to 60. When the CPU is switched to run mode, the level set in the PLR is 63 (the 3 additional levels are used for traps, decribed later in this chapter).
If during program execution one of the 61 interrupts is activated, a number is generated, encoded on 6 bits. This level is compared with the level in the PLR register and if the interrupt level is lower, i.e. of higher priority, and the CPU is in the 'Enable Interrupts' state, the running program or routine is interrupted and the new level is stored in the PLR.
In 'Enable Interrupts' state, the state of the interrupt lines is sampled during each instruction and the interrupt signal is tested at the end of each instruction (and after each step during execution of character string handling instructions and move instructions).

In the P854M, 32 interrupt levels are connected to single interrupt lines. The encoding logic, which generates the 6 bit interrupt level number, is located on the CPU and MIOP cards as follows:

interrupt lines 0 - 3 : internal interrupts on C7E2 card

interrupt lines 4 - 7 : external interrupts on C7E2 card

interrupt lines 8 - 15 : external interrupts on C7E1 card

interrupt lines 15- 31 : external interrupts on MIOP card

In addition to these 32 standard interrupt lines, anadditional 29 levels can be used.

The external interrupt signals must be encoded externally (e.g. on a control unit card) and sent to the CPU either as paralellel coded interrupts (via three interrupt lines and one interrupt group select line) or as serial interrupts via the UPL bus.

SOFTWARE INTERRUPT ACTION

Interrupt action is carried out in two distinct parts:

1. the initial hardware action
2. the programmed software action

Together these actions must ensure that the correct level of program is entered, and that sufficient information with respect to the interrupted level is kept safely so as to enable this level to be restarted correctly once the interrupt has been serviced.

Interrupt Addresses

Associated with each interrupt level is a fixed and reserved word in memory, locations /0 - /78 being used for levels 0 - 60. These locations, referred to as interrupt locations, are addressed from the decoded priority level, and should contain the start address of the software routine associated with the interrupt.

Stack Handling

The information required to restart an interrupted program (i.e. the contentsof the instruction counter and the program status word) is stored in the system stack immediately after a program has been interrupted.

Register A15 is used by the system as a stack pointer to address items in the stack. The system loads the stack's address in A15; each time A15 is used for addressing purposes the contents of A15 are automatically decremented by 2 to point to the next free location.

The stack is used as a last-in/first-out area and is filled from the higher addresses towards the lower addresses. To prevent overwriting of any reserved area at the beginnnng of memory (e.g. interrupt locations) an interrupt is generated when the stack pointer is < /100.

Interrupt Action

As soon as an interrupt is validated the following actions take place:

- the contents of the P register (instruction counter) are stored in the memory location pointed to by A15, the stack pointer
- the contents of A15 are decremented by 2
- the contents of the Program Status Word are stored in the memory stack
- the contents of A15 are decremented by 2
- the CPU is switched to Inhibit mode and system mode
- the priority level of the interrupt is stored in the PLR register
- stack overflow is checked
- the address contained in the interrupt location is loaded into the P register
- the interrupt routine is started

Interrupt Routines

An interrupt routine must at least contain instructions to perform the following actions:

- reset the interrupt that activated the interrupt routine (e.g. by means of an RIT, SST, INR or OTR instruction)
- service the interrupt
- return to the interrupted level by means of an RTN instruction

It may be required that the contents of registers which will be used by the interrupt routine are saved. If so, the system stack or a separate save area may be used for this purpose.
As the interrupt sequence puts the CPU in Inhibit mode, no other interrupts are accepted unless the user puts the CPU in Enable mode by means of the ENB instruction. If the CPU is put in Enable mode before the contents of registers have been saved, any other interrupt routine of a higher priority level should store the contents of registers it intends to use. The order in which storing of registers and the enabling of interrupts is carried out is a matter for the user to decide with reference to the specified requirements of the overall program.
Before a return is made to the originally interrupted level, any working registers which were saved must be restored either from the system stack or from the user's save area.

The last instruction in the interrupt routine must be an RTN instruction specifying register A15 as stack pointer. This instruction performs the following actions:
- register A15 is incremented by 2, to point to the last stack entry
- the Program Status Word of the interrupted program is restored from the stack
- register A15 is incremented by 2
- the P register contents of the interrupted program are restored from the stack

Before execution of the interrupted program is resumed, any outstanding interrupts are checked for priority against the value in the Priority Level Register (part of the PSW). If a higher level interrupt exists, then this is serviced in the manner just explained. If no higher level interrupt exists then the interrupted program is restarted.

TRAPS

A trap is an escape facility, which is activated when one of the following errors occur:
- memory access fault
- not implemented 'D' or 'F' format instruction
- invalid instruction

The trap action is activated regardless of the current priority level or CPU mode (Inhibit/Enable mode, System/User mode). It performs the following actions:
- P and PSW registers are stored in the system stack in the same way as after an interrupt
- the CPU is switched to Inhibit mode and System mode, but the Priority Level Register remains uchanged
- an indirect branch is performed via one of the trap locations /7A, /7C or /7E, depending on the error that activated the trap.

Memory Access Fault

The memory access fault trap is activated by a MMU page fault, i.e. after an attempt to access a wrong or missing page (in user mode), or an attempt to write in a read only page. Apart from the P and PSW register, a third parameter is stored in the system stack. It contains in bits 4-7 the logical page number, all other bits are 0.
The trap location address for the memory access fault is /7A.

Not Implemented Instruction

This trap is activated by any attempt to execute a not implemented instructionin 'D' format or 'F' format (i.e. T8 instruction with OPC=1101 or 1111, for details see chapter 7).
The trap location address is /7C; the associated routine can be used for software simulation of the instruction.

Invalid Instruction

The invalid instruction trap is activated on detection of the following errors:
- an attempt to execute a non-existing instruction
- an attempt to execute a privileged instruction (i.e. an instruction which is restricted to system mode) in user mode
- system stack access in user mode

The trap location address is /7E.

INTERRUPT AND TRAP LOCATION TABLE

The interrupt and trap locations occupy the first 64 words of memory. the table below shows the lay-out of this memory area.

The first 4 interrupt locations are reserved for the standard internal interrupt:

- level 0 : power failure / automatic restart
- level 1 : link to monitor (LKM)
- level 2 : real time clock
- level 3 : control panel

level	address	contents	
0	/00	address of power failure routine	standard
1	/02	address of link to monitor routine	
2	/04	address of real time clock routine	
3	/06	address of control panel routine	
4	/08	address of ext. interrupt routine	machine dependant
5	/0A	address of ext. interrupt routine	
59	/76	address of ext. interrupt routine	
60	/78	address of ext. interrupt routine	
	/7A	page fault trap routine	standard
	/7C	'D' or 'F' format trap routine	
	/7E	invalid instruction trap routine	

9 Input / Output

Data transfers within the system may take place between a master and another master or slave unit, or between two slave units under control of a master. All data transfers take place via the UPL Bus, in the form of parallel 16-bit word or 8-bit character transfers. Figure 9.1 shows a diagramatic layout of the units concerned with data transfers and the associated transfer channels.

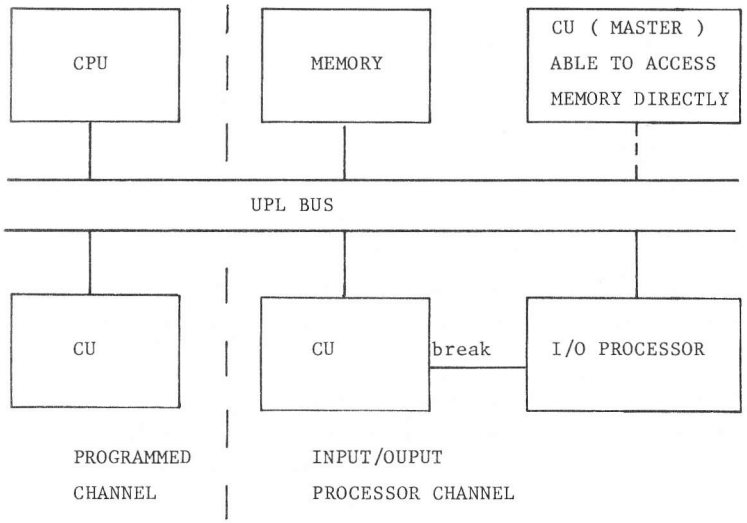


Figure 9.1 Data Transfer Channels and Units

The exchange paths used by the units concerned with data transfers are:

- CPU - control unit
- CPU - external register
- memory - slave unit (under I/O Processor control)
- memory - master unit

The two standard transfer channels - as shown in Figure 9.1 - are the Programmed Channel and the Input/Output Processor Channel.
The Programmed Channel uses the CPU - control unit path to transfer data; in these transfers each word or character exchange requires a separate input/output instruction.
The Input/Output Processor Channel transfers data in blocks between a control unit and memory. Once the CPU has initialized the exchange, the actual data transfer is carried out under control of the I/O Processor hardware, independant from the CPU.

In the P854M two I/O Processors are available on the MIOP board, each with 8 subchannels. Each subchannel can transfer data independently; the priority of each subchannel is prewired at system installation time by means of break connections and corresponding sets of U-links on control unit cards.
Up to 6 external I/O Processors can be connected, each providing 8 independant subchannels.

Data transfers via an I/O Processor use three exchange paths:

- CPU-external register — to set up the transfer parameters
- CPU-control unit — to start or stop the transfer and check the status of the device
- memory-slave unit — to transfer data between memory and control unit via the IOP buffer

In addition to these channels two other modes of operation are possible, each using one of the available exchane paths. Direct access to memory is possible using the memory-master path and transfers between internal and external registers are possible using the CPU-external register path.

PROGRAMMED CHANNEL

The Programmed Channel is the basic transfer channel: apart from providing an exchange facility between a control unit and the CPU it also provides the initialization path between the CPU and the I/O Processor. In all actions concerning the Programmed Channel the CPU is the controlling master.

Data are transferred by input/output instructions and external register instructions in a program and each word or character to be exchanged requires two separate instructions. Apart from the instructions carrying out an exchange, instructions are available to start and stop a device and to check the status of a device. In practice program loops are used to transfer blocks of data.
Two possible modes of operation exist when using the Programmed Channel for data transfers:

- wait mode
- interrupt mode

Wait Mode

This is the simplest but slowest form of transfer and is in most cases not used. Each word or character is exchanged separately and the program is held up in a waiting loop between individual exchanges. In this mode the maximum transfer rate obtainable depends on the time taken to execute the necessary program loop, or the time taken by the device to execute a single exchange, whichever is the slowest.

Interrupt Mode

In interrupt mode data transfers via the Programmed Channel are carried out without the use of time-consuming waiting loops. Each word or character is still exchanged separately, but the input/output instructions form part of an interrupt routine. This means that execution of the main program can continue during the exchange between device and control unit; when the control unit is ready to for the exchange with the CPU an interrupt is raised and the main program is stopped. The interrupt routine carries out the exchange between control unit and CPU, after which the main program can continue until the next interrupt indicates that the control unit is ready for an exchange.
This sequence can be continued until either the transfer is completed or the main program needs the transferred data.

Commands and Responses

The following instructions are used to control transfers via the Programmed Channel:

- CIO Start Start Input/Output
- CIO Stop Stop Input/Output
- INR Input to Register
- OTR Output from Register
- SST Send Status
- TST Test Status

INPUT/OUTPUT PROCESSOR CHANNELS

The Input/Output processor channels provide a fast method of block transfer between the system's memory and device control units.
In P854M systems 2 I/O Processors are mounted on the MIOP board, with 8 subchannels each. Up to 6 external I/O Processors , each controlling 8 subchannels, can be connected to the P854M.
The maximum transfer rate of I/O Processors is 400k words/second.

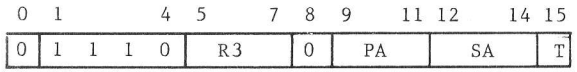
The subchannels of an I/O Processor replace the normal instruction sequence of a Programmed Channel transfer with a hardware sequence to carry out each exchange. A data path is provided between the device control unit and memory via the UPL Bus. The interrupt generated for Programmed Channel transfersis replaced by a Break signal wired directly to the I/O Processor's internal priority system and is used to initiate each harware exchange sequence.

Organization

The priority of a device connected to an I/O Processor depends on the priority of the I/O Processor (i.e. its position in the chain of masters) and the priority of the subchannel, selected by the wiring of break lines.
The subchannels have priorities within the I/O Processor from 0 to 7 (I/OProcessor A), 8 to 15 (I/O Processor B) or 0 to 7 (external I/O Processors).

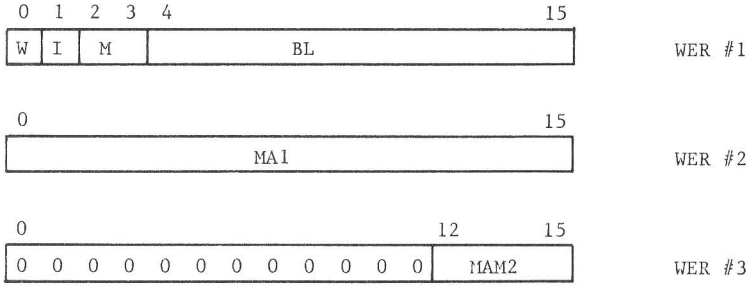
Each I/O Processor contains three 16-bit working registers, which must be loaded with three control words containing transfer parameters. The control words specify: character or word mode, input or output, block length and the first address of the block.
The maximum block length is 4k words or characters.

The control words are loaded into the I/O Processor registers by means of Write External Register (WER) instructions, whose format is:



R3 : number of register containing control word contents
PA : I/O Processor address (for internal I/O Processor: 0 or 1)
SA : subchannel address
T : 0= 1st control word
1= 2nd/3rd control word
Note: the value in bits 9 to 14 is also the 6-bit control unit address

The format of the three control words sent by the three WER instructions is:



W : 0 = character mode
1 = word mode
I : 0 = input (to memory)
1 = output (from memory)
M : 2 most significant bits of memory addresses between 32k and 128k (used for software compatibility reasons, these bits are Overwritten if the 3rd control word is used)

BL : block length, expressed in words or characters, depending on the value of bit 0. The maximum value is $2^{12}-1$; if BL=0, block length is 2^{12}

MA1: least significant 16 bits of memory address; in character mode bit 15=0 for a left hand character and bit 15=1 for a right hand character

MA2: most significant 4 bits of memory address
(this control word is not required for systems <32k)

The order in which the control words are sent is fixed (i.e. the order as described above).

Operation

Three separate control paths are used during an I/O Processor transfer:

1. CPU to I/O Processor

   The CPU is master of the exchange. WER instructions are used to set up the control words, a RER (Read External Register) instruction may be used to check the result of the transfer.

2. CPU to Control Unit

   The CPU is master of the exchange and controls the activation/deactivation or requests the status of a device connected to the I/O Processor. The relevant instructions are sent via the Programmed Channel.

3. I/O Processor to Memory and Control Unit

   The I/O Processor is the master of the exchange taking place between memory and the control unit. The exchange is initiated after the reception of a Break signal and after the I/O Processor has requested and has been granted a bus cycle. The control words are updated (address incremented, length decremented).
   When the last exchange of a transfer has taken place, an interrupt is generated by the control unit. In the associated interrupt routine the correct execution of the transfer can be checked by testing the block length.

DIRECT MEMORY ACCESS

The Direct Memory Access (DMA) channel manages data transfers directly between memory and a single high-speed control unit. The DMA channel hardware is integrated in the control unit logic.
Transfers of data do not use CPU registers and there is no need for program control except for starting the exchange and testing the status after completion.
At the beginning of a transfer, the program uses WER instructions to load transfer parameters into DMA registers, in the same way as described for I/O Processors. The DMA logic then provides all the bus timing signals to control the data transfer between memory and the control unit. The DMA logic also updates the transfer parameters and detects completion of the transfer. The data transfer is ended with an SST instruction and status transfer between the control unit and CPU.

CPU/EXTERNAL REGISTER TRANSFER

The use of exchanges between the CPU and external registers to set up an input/output processor within a standard system has been explained already. Exchanges may also be carried out witn non-standard control ubnits to enable transfers between the CPU and an external register in a control unit for specialized input/output systems. Such systems would operate in a similar manner to the programmed channel and may or may not use the interrupt system to control the timing of exchanges.
Up to 224 external registers may be addressed by the system, for transfers in either direction.

10 Control Panels

Two types of control panel are available:
- the HHCP (Hand Held Control Panel)
- the FRCP (Full Refreshed Control Panel)

HAND HELD CONTROL PANEL

The Hand Held Control Panel is shown in Figure 10.1.
The panel is 120 mm hign and 78 mm wide; it contains data and function buttons, switches, and a single four-digit display.

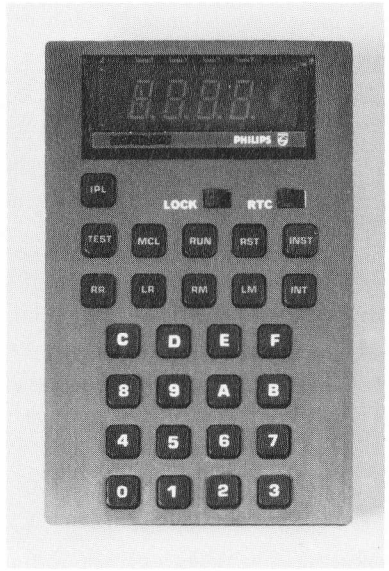


Figure 10.1 Hand Held Control Panel

The functions of the switches, buttons and displays are:

Display

The four-digit display is used to display either data or addresses. The control panel has a buffer for up to five digits, but only the four least significant digits are displayed. The most siginificant digit can be displayed by rotating the displayed data (see descriptiom of MCL button).
Data sent from the CPU to the panel are displayed with a dot in the lower right-hand corner of each digit window.

Data Buttons

The 16 data buttons, marked 0-9 and A-F, are used for entering data in hexadecimal format. Data are entered digit by digit, the most significant digit first. Each digit is displayed in the rightmost position of the display after it has been entered, and is shifted to the left when another digit is entered. Any number of digits can be entered, but only the last four (including leading zeroes) are retained and displayed.
The '0' data button can be used as 'shift key' to extend the function of somefunction buttons. If the '0' data button is pushed and held until the function button has been pushed and released, the extended function of the function button is sent to the CPU.

IPL Initial Program Loader

The IPL button starts the initial program loading procedure, i.e. the system loads the IPL program via device and channel specified on the data buttons.

LOCK Lock Switch

The LOCK switch inhibits all panel functions, except for the displays and indicators.

RTC Real Time Clock

The RTC switch inhibits the interrupts from the real time clock.

TEST Display Test / Microdiagnositc Test

The TEST button either tests the panel by lighting all display segments (TEST button only) or starts the second part of the microdiagnostic test routine ('0' & TEST button).

MCL Master Clear

The MCL button either raises a master clear signal throughout the system, causing a general reset of all system logic (MCL button only) or rotates the displayed data to the left, so that the most significant digit appears in the least significant digit position ('0' & MCL button).

RUN Run Mode

The RUN button switches the CPU to run mode, i.e. the CPU starts operating at the instruction addressed from the P register.

RST Read Status / Preset Access

The RST button either displays the contents of the Program Status Word (RST button only) or instructs the CPU to stop when a previously selected memory address is accessed ('0' & RST button).

INST Single Instruction Mode / Preset Write

The INST button either causes the CPU to execute a single instruction addressed from the P register (INST button only) or instructs the CPU to stop when a write operation occurs at a previously selected memory address ('0' & INST button).

RR Read Register / Read Preset Address

The RR button displays either the contents of a previously selected register (RR button only) or the address on which a 'Stop on preset address' occurred ('0' & RR button).

LR Load Register / Load Address

The LR button loads previously selected data (shown on the display) either into a previously selected register (LR button only) or into the Memory Address register contained in the panel ('0' & LR button).

RM Read Memory

The RM button displays the contents of a memory loacation, addressed from either the P register (RM button only) or from the Memory Address register ('0' & RM button).

LM Load Memory

The LM button loads data into the memory location addressed either from the P register (LM button only) or from the Memory Address register ('0' & LM button).

INT Interrupt / Preset Off

The INT button either raises a control panel interrupt (INT button only) or resets the 'Stop on preset address' function ('0' & INT button).

FULL REFRESHED CONTROL PANEL

The Full Refreshed Control Panel consists of a panel block, mounted in a 4U (177.8 mm) high front panel or in a separate table top box.
The panel block is 180 mm wide and 130 mm high. Figure 10.2 shows the panel block.

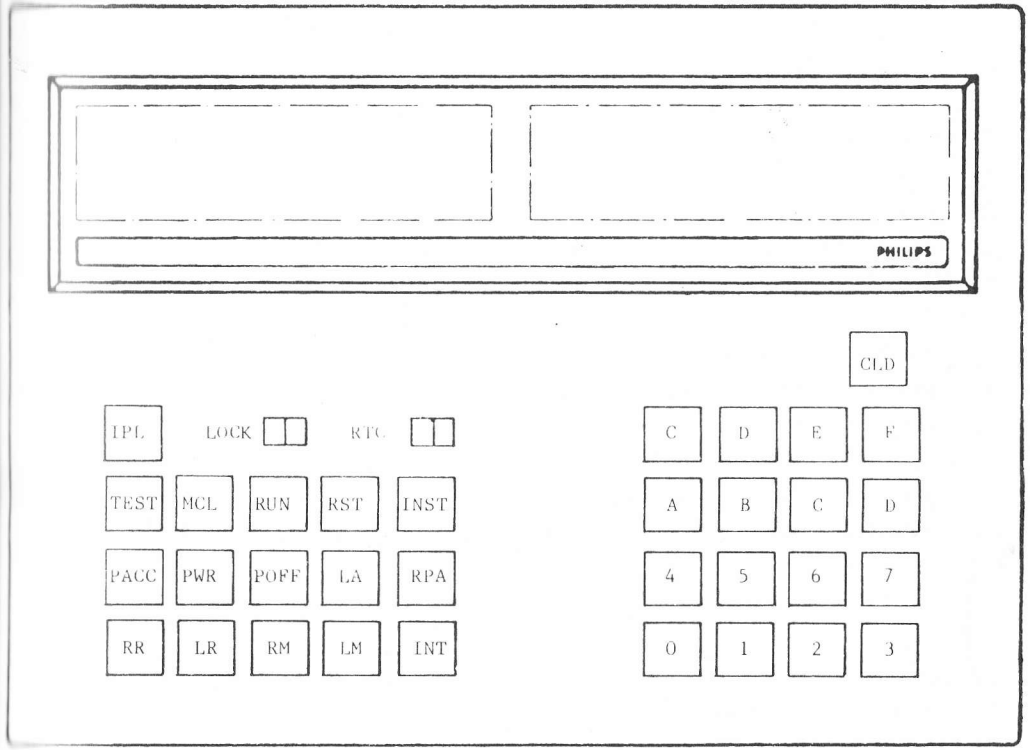


Fig. 10.2 FRCP Panel Block

The panel can be used in its normal control panel function, or in calculator mode to perform hexadecimal addition and/or subtraction.

The facilities available on the panel are:

- the ON/OFF key switch on the front panel
- the switches, pushbuttons and displays on the panel block

ON/OFF Key

This three position key operated switch has the positions:

OFF - power off

ON - power on

MAINT - power off; memory voltages maintained by battery option (if fitted)

The functions of the switches, indicators and displays on the panel block are:

Data Display

The six digit hexadecimal display on the right part of the panel block is used to display data keyed in on the data buttons, or data read from memory or from a register.Data read from memory or from a register are displayed with a small zero in the upper half of the first or first two digit positions.
When the 'Stop on preset address' function is operative, data sent from CPU to panel are displayed with a P in the first digit position.When the CPU is running, 'run' is displayed in lower case on the data display;
if a preset address has been specified, the indication 'run' is followed by a P.In calculator mode, the data display shows the six least significant digits of operands or results.

Address Display

The six digit hexadecimal display on the left part of the panel block is used to display a memory address or a register number, or to display the two most significant digits in calculator mode.

CLD Clear Data

The CLD button clears the data display, and cancels the displayed data.

Data Buttons

The 16 data buttons, marked 0-9 and A-F, are used for entering data in hexadecimal format. Data are entered digit by digit, the most significant digit first. Each digit is displayed in the rightmost position of the display after it has been entered, and is shifted to the left when another digit is entered. Any number of digits can be entered, but only the last six (including leading zeroes) are retained and displayed.
The '0' data button can be used as 'shift key' to extend the function of the LM, RM, TEST and INST buttons. If the '0' data button is pushed and held until the function button has been pushed and released, the extended function of the function button is sent to the CPU.

IPL Initial Program Loader

The IPL button starts the initial program loading procedure, i.e. the system loads the IPL program via device and channel specified on the data buttons.

LOCK Lock Switch

The LOCK switch inhibits all panel functions, except for the displays and indicators, and enables the automatic restart function.

RTCE Real Time Clock

The RTC switch enables the interrupts from the real time clock.

TEST Panel Test / Microdiagnostic Test

The TEST button is used to test the displays and to start the microdiagnostic test. Pushing only the TEST button causes all segments of all displays to be lit (internal panel test); pushing and releasing the TEST button when the '0' data button is pushed and held, starts the second part of the microdiagnostic test routine.

MCL Master Clear

The MCL button raises a master clear signal throughout the system, causing a general reset of all system logic.

RUN Run Mode

The RUN button switches the CPU to run mode, i.e. the CPU starts operating at the instruction addressed from the P register.

RST Read Status

The RST button displays the contents of the Program Status Word on the data display; the address display is not affected.

INST Single Instruction Mode / Calculator Mode

The INST button causes the CPU to execute the instruction addressed from the P register, and then stop. If the INST button is pushed when the CPU is running, the CPU is stopped and put in the idle state.
Pushing and releasing the INST button when the '0' data button is pushed and held switches the panel to calculator mode; pushing the INST button when the panel is in calculator mode switches it back to normal mode.

PACC Preset Access

The PACC button instructs the CPU to stop when a previously selected memory address (shown on the address display) is accessed.

PWR Preset Write

The PWR button instructs the CPU to stop when a write operation occurs at a previously selected memory address (shown on the address display).

POFF Preset Off

The POFF button cancels the 'Stop on preset address' request from the PACC or PWR button.

LA Load Address

The LA button loads a previously selected address (shown on the data display)into the Memory Address register , and displays the address on the addressdisplay.

RPA Read Preset Address

The RPA button displays the address specified when the PACC or PWR button was pushed after a 'Stop on preset address' has occurred. The address is displayed on the address display, the data display is not affected.

RR Read Register

The RR button displays on the data display the contents of a previously selected register; the register number (entered via the data buttons) is shown on the data display before the RR button is pushed and on the address display after the RR button has been pushed.

LR Load Register

The LR button loads previously selected data (shown on the data display)into a previously selected register (shown on the address display).

RM Read Memory

The RM button displays on the data display the contents of a memory location, addressed from either the Memory Address register (RM button only) or from the P register ('0' & RM button).

LM Load Memory

The LM button loads data into the memory location addressed either from the Memory Address register (LM button only) or from the P register ('0' & LM button).

INT Interrupt

The INT button raises a control panel interrupt.

11 Control Panel Operating Procedures

The control panel operating procedures are described below in two sections:

- the HHCP (Hand Held Control Panel) operating procedures
- the FRCP (Full Refreshed Control Panel) operating procedures

HHCP Operating Procedures

The following operations can be performed from the control panel:

- loading data into a register
- loading data into memory
- reading data from a register
- reading data from memory
- reading the Program Status Word
- loading a program
- executing a single program instruction
- stopping a program at a preset address

In all operations described below, it is assumed that power is supplied to the system, and that the system is switched on but not running.

Load Register

Step 1 Push the data button with the required register number

Step 2 Push the RR button; the register number appears on the display,

Step 3 Enter the data to be loaded on the data buttons

Step 4 Push the LR button; the 4 rightmost digits shown on the display are loaded into the selected register

Load Memory

Data can be loaded into memory in two ways: by addressing the memory location from the P register, or by addressing the memory location from the Memory Address register.

Load Memory via P Register

Step 1 Push data button 0

Step 2 Push the RR button; the current contents of P register appear on the display

Step 3 Enter the required address on the data buttons

Step 4 Push the LR button; the 4 rightmost digits shown on the display arey loaded into the P register

Step 5 Enter the data to be loaded on the data buttons

Step 6 Push the LM button; the displayed data are loaded into the memory location addressed from the P register; the P register is incremented by 2.

To load the same data value into consecutive memory locations, push the LM button as many times as required.

Step 7 To load different data values into consecutive memory locations, go back to step 5

Load Memory via Memory Address Register

Step 1 Enter the required memory address on the data buttons

Step 2 Push and hold data button 0

Step 3 Push and release the LR button, then release data button 0; the specified memory address is loaded into the Memory Address register

Step 4 Enter the data to be loaded on the data buttons

Step 5 Push and hold data button 0

Step 6 Push and release the LM button, then release data button 0; the keyed in data are loaded into the memory location addressed from the Memory Address register and the Memory Address register contents are incremented by 2

To load the same data value into consecutive memory locations, push the LM button as many times as rquired.

Step 7 To load different data values into consecutive memory locations, go back to step 4

Note: The P register contents are not affected.

Read Register

Step 1 Push the data button with the required register number

Step 2 Push the RR button; the contents of the register appear on the display

Read Memory

Data can be read from memory in two ways: by addressing the memory location from the P register, or by addressing the memory location from the Memory Address register contained in the control panel.

Read Memory via P Register

Step 1 Push data button 0

Step 2 Push the RR button; the current contents of P register appear on the display

Step 3 Enter the required address on the data buttons

Step 4 Push the LR button; the address shown on the display is loaded into the P register

Step 5 Push the RM button; the contents of the the memory location addressed from the P register are displayed; the P register is incremented by 2

Step 6 For reading consecutive memory locations, go back to step 5

Read Memory via Memory Address Register

Step 1 Enter the required memory address on the data buttons

Step 2 Push and hold data button 0

Step 3 Push and release the LR button, then release data button 0; the specified memory address is loaded into the Memory Address register

Step 4 Push and hold data button 0

Step 5 Push and release the RM button, then release data button 0; the contents of the memory location addressed from the Memory Address register are displayed and the Memory Address register contents are incremented by 2

Step 6 For reading consecutive memory addresses, go back to step 5

Read Program Status Word

To read the Program Status Word (PSW), push the RST button. The PSW appears on the display; the significant bits of the PSW have the following meaning:

- bit 0-5 priority level register
- bit 6-7 condition register
- bit 9 enable interrupt indicator
- bit 15 user mode indicator

Loading a Program

Program loading is carried out in three separate stages:

- the operator loads the bootstrap
- the bootstrap loads the Initial Program Loaded (IPL)
- the IPL loads a system or user program

The bootstrap is a basic loading program used to initialize the system and to load more sophisticated loader programs. The bootstrap is contained in a Read Only Memory (ROM).

The IPL is written in machine code, and is loaded by the bootstrap from a peripheral device.

The IPL loads object code clusters into memory, and contains error reporting facilities.

IPL Parameters

The operator must specify the device and channel used for loading the IPL, by means of parameters entered via the data buttons. The parameter values can be found in Appendix A.

IPL Loading

The IPL loading procedure is:

Step 1 Prepare the device from which the IPL is to be loaded

Step 2 Enter the required parameters on the data buttons

Step 3 Push the IPL button

After the IPL has been loaded into memory, it starts loading a system or user program automatically from the same device.

Single Instruction Mode

For debugging purposes, a sequence of instructions can be executed in single instruction mode as follows (if it is not necessary to load a new start address, e.g. to continue execution of a stopped program, steps 1 to 4 can be omitted):

Step 1 Push data button 0 to select the P register

Step 2 Push the RR button

Step 3 Enter the address at which execution must start on the data buttons

Step 4 Push the LR button to load the start address into the P register

Step 5 Push the INST button to execute the instruction addressed from the P register; the P register is incremented by 2

Step 6 If required, push the INST button to execute the next instruction

Stop on Preset Address

A sequence of instructions can be executed, and then stopped at a preset address as follows:

Step 1 Enter the address at which the program must stop on the data buttons

Step 2 Push and hold data button 0

Step 3 Push and release either the RST or INST button: '0'& RST stops the program whenever the specified address is accessed, '0' & INST stops the program only on write access at the specified address

Step 4 Push the Run button to start execution

FULL REFRESHED CONTROL PANEL OPERATING PROCEDURES

The following operations can be performed from the control panel:

- loading data into a register
- loading data into memory
- reading data from a register
- reading data from memory
- reading the Program Status Word
- loading a program
- executing a single program instruction
- stopping a program at a preset address
- hexadecimal caculations (in calculator mode)

In all operations described below, it is assumed that the safety key switch is in the ON position, that power is supplied to the system, and that the system is not running.

Load Register

Step 1 Push the data button with the required register number (if more than one digit is entered, only the least significant one is used)

Step 2 Push the RR button; the register number appears on the address display, its current contents on the data display

Step 3 Enter the data to be loaded on the data buttons

Step 4 Push the LR button; the 4 rightmost digits shown on the data display are loaded into the selected register, the register number remains displayed on the address display

Load Memory

Data can be loaded into memory in two ways: by addressing the memory location from the Memory Address register or by addressing the memory location from the P register (via the P register only the first 32k words of memory can be addressed).

Load Memory via Memory Address Register

Step 1 Enter the required memory address on the data buttons

Step 2 Push the LA button; the address is loaded into the Memory Address register and is displayed on the address display

Step 3 Enter the data to be loaded on the data buttons

Step 4 Push the LM button; the 4 rightmost digits shown on the data display are loaded into the specified address, and the memory address is incremented by 2

To load the same data value into consecutive memory locations, push the LM button as many times as required.

Step 5 To load different data values into consecutive memory locations, go back to step 3

Note: The P regiser contents are not affected.

Load Memory via P Register

Step 1 Push data button 0

Step 2 Push the RR button; 0 appears on the address display and the current contents of P register appear on the data display

Step 3 Enter the required address on the data buttons

Step 4 Push the LR button; the 4 rightmost digits shown on the data display are loaded into the P register

Step 5 Enter the data to be loaded on the data buttons

Step 6 Push and hold data button 0

Step 7 Push and release the LM button; the 4 rightmost digits shown on the data displayare loaded into the specified address, and the P register contents are incremented by 2

To load the same data value into consecutive memory locations, push the LM button as many times as required.

Step 8 Release data button 0

Step 9 To load different data values into consecutive memory locations, go back to step 5

Read Register

Step 1 Push the data button with the required register number

Step 2 Push the RR button; the register number appears on the address display, the register contents appear on the data display

Read Memory

Data can be read from memory in two ways: by addressing the memory location from the Memory Address register contained in the control panel, or by addressing the memory location from the P register (via the P register only the first 32k words of memory can be addressed).

Read Memory via Memory Address Register

Step 1 Enter the required memory address on the data buttons

Step 2 Push the LA button; the address is loaded into the Memory Address register and is displayed on the address display

Step 3 Push the RM button; the contents of the selected address are displayed on the data display, and the memory address is incremented by 2

Step 4 For reading consecutive memory locations, go back to step 3

Note: The P register contents are not affected.

Read Memory via P Register

Step 1 Push data button 0

Step 2 Push the RR button; 0 appears on the address display and the current contents of P register appear on the data display

Step 3 Enter the required address on the data buttons

Step 4 Push the LR button; the 4 rightmost digits shown on the data display are loaded into the P register

Step 5 Push and hold data button 0

Step 6 Push and release the RM button; The contents of the selected address are displayed on the data display, and the P register contents are incremented by 2

Step 7 For reading consecutive memory locations, go back to step 6

Read Program Status Word

To read the Program Status Word (PSW), push the RST button. The PSW is displayed on the data display, the address display is reset to 00000.The significant bits of the PSW have the following meaning:

- bit 0-5 priority level register
- bit 6-7 condition register
- bit 9 enable interrupt indicator
- bit 13 extended system mode indicator
- bit 15 user mode indicator

Loading a Program

Program loading is carried out in three separate stages:

- the operator loads the bootstrap
- the bootstrap loads the Initial Program Loader (IPL)
- the IPL loads a system or user program

The bootstrap is a basic loading program used to initialize the system and to load more sophisticated loader programs. The bootstrap is contained in a Read Only Memory (ROM).

The IPL is written in machine code, and is loaded by the bootstrap from a peripheral device.

The IPL loads object code clusters into memory, and contains error reporting facilities.

IPL Parameters

The operator must specify the device and channel used for loading the IPL, by setting parameters on the data switches. The parameter values can be found in Appendix A.

IPL Loading

The IPL loading procedure is:

Step 1 Prepare the device from which the IPL is to be loaded

Step 2 Enter the required parameters on the data buttons

Step 3 Push the IPL button

After the IPL has been loaded into memory, it starts loading a system or user program automatically from the same device.

Single Instruction Mode

For debugging purposes, a sequence of instructions can be executed in single instruction mode as follows (if it is not necessary to load a new start address, e.g. to continue execution of a stopped program, steps 1 to 4 can be omitted):

Step 1 Push data button 0 to select the P register

Step 2 Push the RR button

Step 3 Enter the address at which execution must start on the data buttons

Step 4 Push the LR button to load the start address into the P register

Step 5 Push the INST button to execute the instruction addressed from the P register; the P register is incremented by 2

Step 6 If required, push the INST button to execute the next instruction

Each time the INST button is pushed, the address of the next instruction appears on the address display, and its contents on the data display.

Stop on Preset Address

A sequence of instructions can be executed, and then stopped at a preset address as follows:

Step 1 Enter the address at which the program must stop on the data buttons

Step 2 Push either the PACC or the PWR button: PACC stops the program when the specified address is accessed, PWR stops the program only if write access occurs at the specified address

Step 3 Push the Run button to start execution.
The address selected by PACC or PWR is shown on the address display, on the data display 'P' is shown next to the 'run' indication.

Calculator Mode

In calculator mode, hexadecimal additions and/or subtractions can be carried out on the control panel.

Pushing the INST button while the '0' data button is pushed and held switches the panel to calculator mode, indicated by a 'C' in the most significant digit position of the address display.

The following buttons may be used in calculator mode:

button	function	
MCL	CA	clear all: both displays go blank
CLD	CE	clear entry: the previous entry is cancelled and the display shows C 00 000000
RR	+	addition
RM	-	subtraction
INT	=	result
INST		return to 'normal' mode

Operands are entered via the data buttons; the display shows either the operand after data have been entered or the result after the +, - or = function button has been pushed. The maximum capacity of the displays in calculator mode is 8 digits: the 6 least significant digits appear on the data display, the 2 most significant digits in the least significant positions of the address display. Operations may involve more than two operands, e.g.

6+3-5+2+F

If the + or - function button is pushed repeatedly, the associated function is performed on the last result and the previous operand:

operand	operator	display
6	+	6
3		3
	+	9
	+	C
	=	F

12 Additional Standard Features

Apart from the main facilities already covered, the following additional features are available:

- power failure/automatic restart
- real time clock
- integrated V24/V28 serial control unit
- microdiagnostics
- detection of privileged instructions

POWER FAILURE / AUTOMATIC RESTART

The power failure / automatic restart facility enables the CPU to detect power breaks in time to terminate processing, save vital program parameters, and to restart and resume processing correctly after power has been restored.

Power Failure

Apart from separate peripherals, all power supplies within the system have detection units which activate a Power Failure (PF) signal whenever a main's power break longer than 40 ms occurs.
If the control panel Lock switch is set to Lock, a PF signal raises an interrupt at level 0 (i.e. highest priority), which activates the associated power failure / automatic restart interrupt routine.The contents of P and PSW registers are saved automatically; the 'power failure' part of the interrupt routine can be used to save the contents of general purpose registers. The routine must contain an RIT instruction to reset the interrupt, and must end with a Halt (HLT) instruction to put the CPU in the idle state.
As Power Failure is set at least 2 ms before the DC voltages drop below the acceptable level, the saving routine should not take longer than 2 ms.

Automatic Restart

If the control panel Lock switch is in the Lock position, a power failure interrupt is raised when power is restored. The interrupt switches the CPU to run mode, and activates the power failure / automatic restart interrupt routine. The 'automatic restart' part of this routine must restore all registers which were saved in the 'power failure' part, and it must contain an RIT instruction to reset the interrupt.At the end of the interrupt routine, control is returned to the program which was interrupted by the power failure.

REAL TIME CLOCK

The real time clock generates an interrupt every 20 ms for 50 cps supplies, or every 16.67 ms for 60 cps supplies. It can be switched on and off by means of the control panel RTC switch.
The real time clock interrupt, connected to interrupt level 2, must be cleared by a Reset Internal Interrupt (RIT) instruction. The associated interrupt routine may be used as required within the system.

MICRODIAGNOSTICS

The P854M contains an automatic testing feature in the form of a microprogrammed diagnostic built into the CPU logic. Succesful running of the tests indicates that sufficient parts of the CPU function for loading of test programs.

Test Procedures

The microdiagnostic test consists of two test routines: the first routine is executed automatically when the CPU is switched on and the control panel is not locked, the second routine is started when the '0' and TEST buttons on the control panel are pushed.
The first test routine tests the main part of the CPU and the control panel. If the test has been executed satisfactory, code /FFFC is displayed.

The second test routine tests the other parts of the CPU, the memory up to 32k words, the dialogue between the CPU and the integrated serial control unit, the Memory Management Unit, and the built-in I/O processor.
The results of the test are displayed as follows:

/0001	CPU error
/0002	bus or control unit error
/xx03	bus error , or memory error (xx: 8 most significant address bits)
/yy04	o.k. (yy: 8 most significant bits of highest memory address)
/0010	MMU error or I/O processor error

Information about the memory word causing an error can be read from register A1 (address) and register A2 (contents of address).

INTEGRATED V24 SERIAL CONTROL UNIT

A V24/V28 Serial Control Unit is available within the system, mounted on the CPU board. This control unit allows connection of an asynchronous peripheral device (e.g. ASR, PER3100, P817 display terminal) as I/O console.
The data exchange is character oriented, and has the following features:

- 7 or 8 data bits, selectable by means of the CIO instruction
- odd, even or no parity, selectable by means of the CIO instruction
- 2 stop bits in transmitting mode, 1 or 2 stop bits accepted in receiving mode
- transmission speeds of 110, 600, 1200, 2400, 4800 or 9600 bps, selectable by straps
- echo mode or no echo mode, selectable by means of the CIO instruction

Interface

The control unit operates via the programmed channel. The control unit device address must be /10.
The interface between control unit and device must meet CCITT V24 recommendations. Figure 12.1 shows the control unit interfaces.

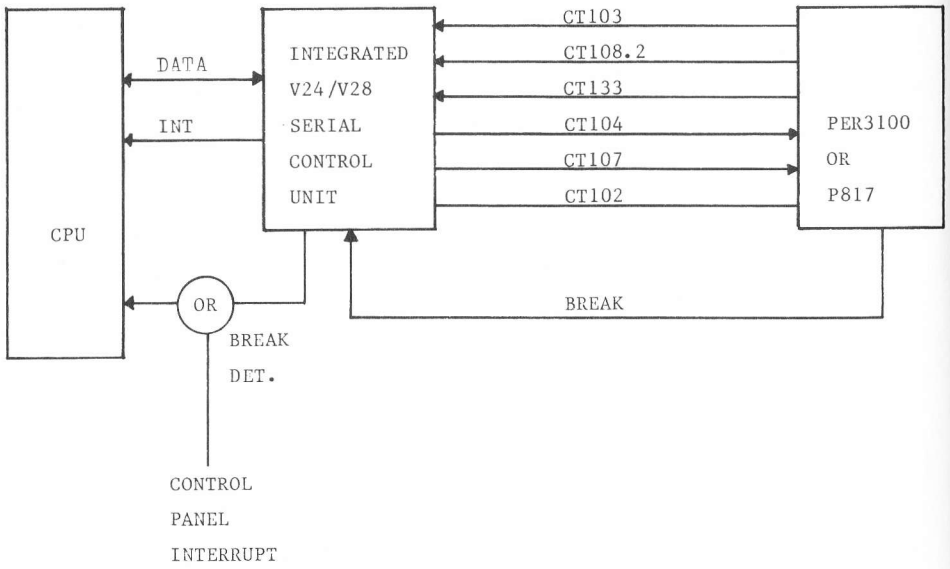


Figure 12.1 Integrated Serial Control Unit

Break Feature

The Break key on the peripheral's keyboard has the same function as the control panel INT button. The control unit can detect the Break character in any mode; the associated interrupt is 'ORed' with the control panel interrupt.

Operation

The control unit operates in the same way as other control units, commencing in the inactive state, transferring during the exchange and execute state, and stopping in the wait state. Operation may be either in wait or interrupt mode. Input/output instructions recognized by the control unit are:

- CIO Start input/output
- CIO Stop
- OTR
- INR
- SST

Responses to these instructions are set in the condition register in the normal way.

Figures 12.2 and 12.3 show flowcharts of possible methods of programming the integrated serial control unit.

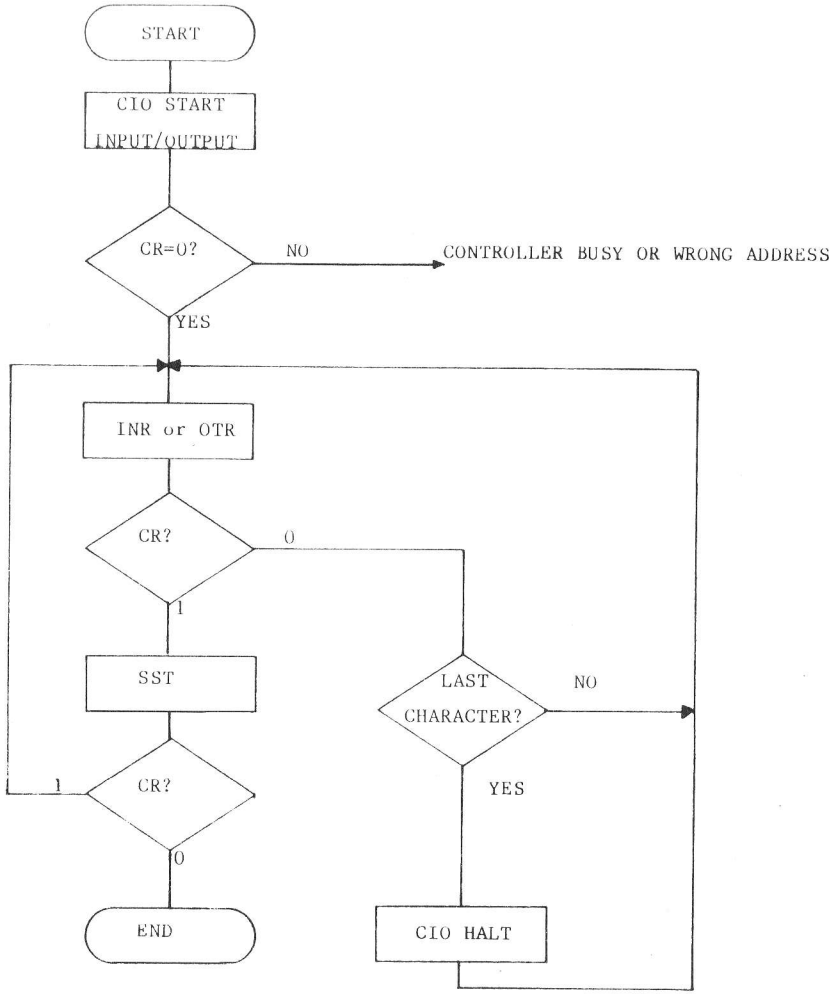


Figure 12.2 Wait Mode

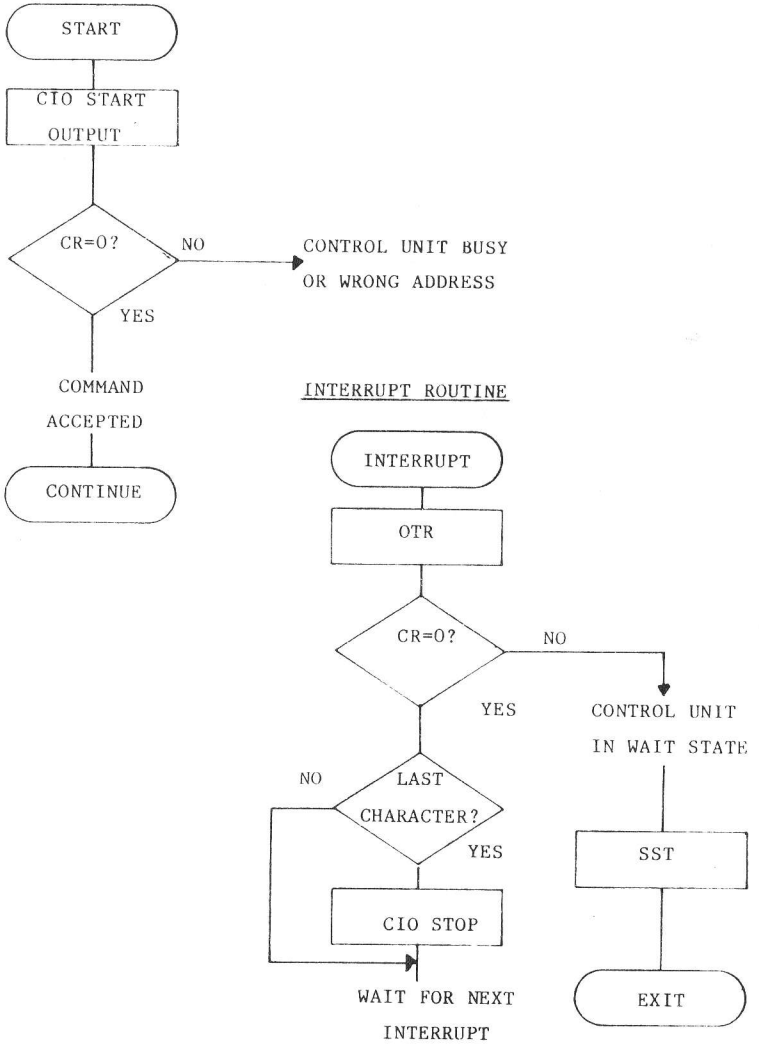


Figure 12.3 Interrupt Mode

DETECTION OF PRIVILEGED INSTRUCTIONS

The CPU may operate in three modes:

- system mode
- extended system mode- user mode

System Mode

In system mode, all available instructions may be executed, including the privileged instructions restricted to this mode (I/O instructions, external transfer instructions, and instructions modifying the CPU state or affecting the system stack). Only the monitor and system programs are executed in this mode.

The MMU remains inactive, except for the translation of addresses in special MMU instructions (Extended Load / Store, Move Table from System to User and Move Table from User to System). No memory protection checks are performed.

Extended System Mode

In extended system mode, all addresses are translated by the MMU. No memory protection checks or privileged instruction checks are performed. For programs running in the first 32k words of memory, the MMU performs a dummy translation.When the CPU is operating in extended system mode, bit 13 of the Program Status Word is set to 1. The CPU is switched to extended system mode by an RTN A15 instruction, if bit 13 of the PSW contents in the stack has been set to 1.

User Mode

In user mode, all addresses are translated by the MMU, and memory protection checks are performed for each memory access. Any attempt to execute one of the privileged instructions activates the invalid instruction trap (trap location /7A).

User programs operating under monitor control are executed in this mode. To carry out an operation which is restricted to system mode, the user program must issue a request to the monitor by means of a Link to Monitor (LKM)instruction.

When the CPU is operating in user mode, bit 15 of the PSW is set to 1.

13 Data Communication and Digital Input / Output

Data Communication

A full range of data communication control units for synchronous and asynchronous transmission is available, making use of the latest LSI technology for increased performance and reliability.
All control units may be plugged in the mounting box or equipment shelf.

The following data communication control units are available:

SALCUZ P845-160	A synchronous and asynchronous line control unit, handling one half or full duplex line, at possible operating speeds of 110, 200, 300, 600 1200, 2400, 4800 or 9600 bits/sec (asynchronous) or 1200, 2400, 4800, 9600 or 19200 bits/sec (synchronous).
SLCUZ P845-170	A synchronous control unit, handling one half duplex or full duplex line, at operating speeds up to 4800 bits/second (full duplex) or 9600 bits/second (half duplex). Interface according to V24/V28 recommendations.. Facilities include special character recognition, CRC/VRC/LRC generation and check, and remote control of modem loops.
HLCUZ P845-100	Control unit for High Level Data Link procedures, operating in half or full duplex mode, at speeds up to 100,000 bits/sec (inplant) or up to 20,000 bits/sec (outplant). The interface is according to CCITT V24/V28 recommendations; the control unit uses frame structures according to HDLC/SDLC standards.
HLVCUZ P845-110	Control unit for High Level Data Link procedures, operating in half or full duplex mode, at speeds up to 100,000 bits/sec (inplant) or up to 20,000 bits/sec (outplant). The interface is according to CCITT V24/V28 recommendations; the control unit uses frame structures according to HDLC/SDLC standards.

Facilites include modem loop control and automatic address field detection.

ASCU4Z
P845-145 — A multiple serial unit, for 4 half duplex or 2 full duplex lines in local connections.

Remote Loading

When two P800 configurations are connected by means of a data communications link, the remote loading facility enables the user to load the remote P800 configuration with a previously generated system (i.e. monitor and application programs) from a device connected to the local P800 configuration.

The loading operation is started from a Supervisory Panel, connected to an IPLC (Initial Program Loader Control Unit) card in the remote P800. The IPLC card contains one or more IPL programs, held in Read Only Memory, which replace the standard IPL program. Switches on the Supervisory Panel select which IPL will be activated.
In the local P800 configuration, the actual loading operation is performed by a software routine REMLOD, running under monitor control.
In both configurations a HLCUZ card is required for data transmission. The data communication link between the two configurations may be any type which supports the HDLC protocol.
Full details about the remote loading procedure are given in P800M Programmer's Guide 1, Vol. IV (BRTM).

Modular Input/Output System

MIOS (Modular Input/Output System) is an adaptable interface system that can be configured to suit any analog/digital application. Basic MIOS functions include signal acquisition, interrogation, rationalization to computer requirements, and producing output for control, display and record.
MIOS is a separate rack mpunted system, connected to the UPL Bus by means of a bus translator card.

The following modules are available:

Digital I/O

PC 1710/00 — Input board for 32 bits input
PC 1711/00 — Isolated input board for 32 bits input, with opto coupler isolation and display facility
PC 1715/00 — Input/output board for 16 bits input and 16 bits output
PC 1716/00 — Isolated input/output board for 16 bits input and 16 bits output, with opto coupler isolation and display facility
PC 1720/00 — Output board for 32 bits output
PC 1721/00 — Isolated output board for 32 bits input, with opto coupler isolation and display facility

Analog I/O

PC 1740/00 — Input low level board for 8 analog inputs with reed relays
PC 1741/00 — Input high level board for 8 analog inputs with reed relays
PC 1750/00 — Programmable amplifier and analog to digital converter board, with 12 bits ADC
PC 1751/00 — Reference source board, for reference and compensation of the Programmable amplifier
PC 1755/00 — Input fast module and ADC function 16 solid state channel multiplexor with a 12 bit ADC
PC 1760/00 — Output board for 10 bits buffered non-isolated analog output
PC 1761/00 — Isolated output board for 10 bits buffered analog output
PC 1762/00 — Output board for 2 x 10 bits buffered, non-isolated analog outputs

Extension and Exerciser

PC 1701/00 — Bus extension board
PC 1730/00 — Digital exerciser board with switches and display facilities to test the digital interfaces

Data Link Controller

PC 1790/00 — Central coaxial data link controller board, enables connection, up to 1 km, of a MIOS rack
PC 1791/00 — Remote coaxial data link controller, provides the function for remote control to the central processor

14 Cabinets

CABINET

The cabinet is a metal structure containing a standard 19" rack, which is used to hold a mounting boxes, equipment shelves and various peripheral equipment.

MOUNTING BOXES

The mounting boxes contain power supply units, card slots for logic cards, and a back panel printed circuit board which carries the UPL bus signals and the required power connections. A front panel is mounted to the front of the mounting box, with an operator's key switch and provisions for the Hand Held Control Panel, the Full Refreshed Control Panel, or no panel.

Two types of mounting boxes are available:

- the 6U12 mounting box, with 12 card slots and a power supply unit
- the 10U22 mounting box, with 22 card slots (this mounting box requires a separate power supply)

For both mounting boxes, the following configuration rules apply:

- the card slots are numbered from 1 to 12, from left to right when seen from the open side of the mounting box
- cards are inserted vertically, with the component side facing right
- CPU card C7E2 must be plugged into slot 1
- CPU card C7E1 must be plugged into slot 2
- the MIOP card must be plugged into slot 3
- other masters can be plugged into the other slots, in order of system priority as defined by the OKO/OKI chain

The mounting boxes are 6U (266 mm) high. A ventilation unit is standard forthe 10U22 mounting box, for the 6U12 mounting box it is mandatory if mounted in a cabinet, otherwise it is highly recommended. The ventilation units provide forced air cooling for the logic cards and power supply (6U12 mounting box) or for the logic cards only (10U22 mounting box).

The ventilation unit occupies 3U (133 mm) on top of the mounting box; if it is fitted, 1U space must be left free underneath the mounting box to ensure unobstructed air flow.

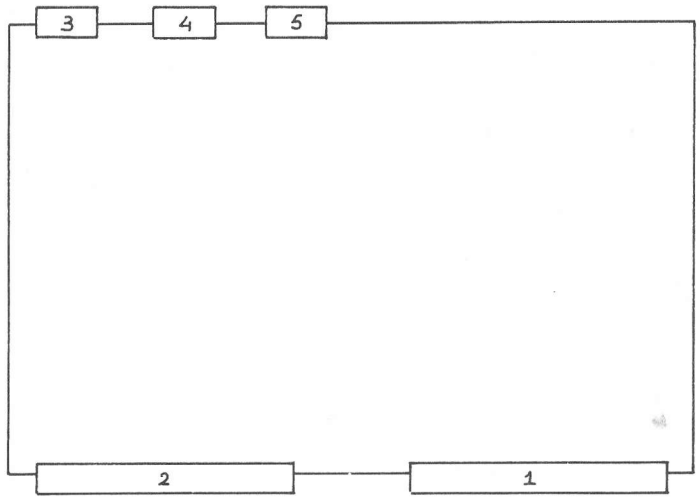
Cards

The standard card used in the P854M is the double Eurocard, measuring 160 * 233.4 mm.

One connector with 3 * 32 pins connects the card to the UPL bus. On the CPU cards and the MIOP card, a second connector with 3 * 32 pins is used for special interconnections; on the C7E2 card 3 connectors on the opposite side of the card are used for CPU interfaces.

Figure 14.1 shows the C7E2 card, with five connectors:

- connector 1 - UPL bus interface
- connector 2 - interconnections to C7E1 and MIOP cards
- connector 3 - remote loading interface
- connector 4 - control panel interface
- connector 5 - V24 serial interface



POWER SUPPLIES

The power supply units are either contained in the mounting box (6U12) or mounted as a separate unit in the cabinet (10U22).

Power Supply Unit for 6U12 Mounting Box

The double power supply unit for the 6U12 mounting box consist of two separate units, each providing power for up to six cards. One unit, acting as master, is connected to the mains and to the operator's key switch, the other unit, acting as slave, receives its mains supply from the master and is switched on and off by the master. The master unit is equipped with a mainsfilter unit.

Both master and slave unit can be equipped with a battery back-up option, which supplies power to the MOS memory cards from the power supply when power is available, or from the batteries when a power failure occurs or when theoperator's key switch is set to MAINT. If the battery back-up option is not fitted, power for the memory cards is derived from the logic voltages.

A double power supply unit can provide the following voltages/currents:

$+5V_L$	2 * 14A
$+12V_L$	2 * 1.6A
$-12V_L$	2 * 0.5A
$+5V_M$	1 or 2 * 1.5A (battery back-up card)
$+12V_M$	1 or 2 * 0.8A (battery back-up card)

The power supply units operate on line voltages of 115V, 220V, or 240V $\pm$10%, single phase with ground or double phase with ground, at either 50 Hz $\pm$2% or 60 Hz $\pm$3%. All units are provided with overvoltage and overcurrent protection.

An optional ventilation unit can be mounted on top of the mounting box to provide forced air cooling for the power supply units and the logic cards.

Power Supply Unit for 10U22 Mounting Box

Power for the 10U22 mounting box is provided by a separate power supply unit, with one or two power supply modules for logic voltages and one or two power supply modules for memory voltages.
The power supply unit fits into the 19" rack behind the RU22 mountingbox. Ventilation is provided by a ventilation unit mounted on top of the power supply unit.

A battery back-up feature supplies power to MOS memory cards during mains power failures, or when the operator key switch is set to MAINT. The battery can maintain power supply to a 512k word memory for 45 minutes and to a 128k word memory for 2 hours 45 minutes.

The power supply units operate on line voltages of 115V, 220V, or 240V ±10%, single phase with ground or double phase with ground, at either 50 Hz ±2% or 60 Hz ±3%. All units are provided with overvoltage and overcurrent protection.

The power supply unit can provide the following voltages/currents:

	one module	two modules	
$+5V_L$	40A	60A	
$+12V_L$	1.5A	3.0A	
$-12V_L$	1.0A	2.0A	
$+5V_M$	2.5A	5.0A	with ventilation unit
	2.0A	4.0A	without ventilation unit
$+12V_M$	1.4A	2.8A	with ventilation unit
	1.0A	2.0A	without ventilation unit

Power Failure

When the system detects a mains failure of more than 10 msec, a Power Failure signal is sent to the CPU to activate the Power Failure interrupt routine.

Real Time Clock

The Real Time Clock circuit uses the a.c. output from the mains transformer to produce a 1 usec pulse at mains frequency (every 20 msec at 50 Hz or every 16.67 msec at 60 Hz). This signal is sent to the CPU to activate the Real Time Clock interrupt routine.

15 Peripheral Equipment

A comprehensive range of standard peripheral equipment is available for use within the system, and where it is required non-standard and customer built devices may be connected either separately or in conjunction with standard equipment.
The standard peripheral devices currently available include:

Input/Output Typewriters

P842-001 PER3100 matrix printer with keyboard, V24 interface
P842-002 PER3100 matrix printer with keyboard, current loop interface

Display Equipment

P817-001 Video display terminal, character oriented
P817-002 Video display terminal, character/block oriented

Punched Tape Equipment

P801-001 Punched tape reader, 333 char/sec
P803-001 Tape punch, 75 char/sec

Card Reader

P806-102 Punched card reader, 300 cards/min

Line Printers

P809-002 Matrix line printer, 200 lines/min
P809-004 Matrix line printer, 400 lines/min

Magnetic Disc Equipment

P824-012 Moving head cartridge disc drive, 2 x 2.7M bytes
P824-014 Moving head cartridge disc drive, 2 x 5.4M bytes
P827-001 Fixed disc drive, 5.9M bytes
P830-006 Flexible disc drive, 250k bytes
P830-015 Flexible disc drive, 250k bytes, with door lock
P830-025 Flexible disc drive, 1M bytes
P830-035 Flexible disc drive, 1M bytes, with door lock

POWER SUPPLIES

The necessary power for all standard peripheral devices is produced by either self-contained power supply units or by a separate unit mounted together with the device in either the basic cabinet or in an equipment shelf.
Power for the associated control unit is derived from the power supply within the mounting box or equipment shelf or from the peripheral's separate power supply.

CONNECTION TO THE SYSTEM

Peripheral devices are connected to the system via a control unit; the control unit may be connected to either the Programmed Channel or an I/O Processor. Transfer rates up to the maximum operating speed of the device and/or the maximum throughput rate of the channel are possible, but either rate may be limited by the other.

CONTROL UNITS

The table on the following page list the standard control units.

Note: O connection to the specified channel is possible but is not supported by standard software
X connection supported by standard software
- connection not possible

Type Number	CU	Channel connection: Prog.Channel	Channel connection: I/O Proc	Int/ Breaks	Remarks
P801-140	PTR	X	O	1	separate CU
P840-141	PTR & PTP	X	O	2	multiple CU
P840-103	CR & LP	O	X	2	multiple CU
P845-140	V24	X	-	1	separate CU
P824-140	Disc	-	X	1	CU for 2 drives
P830-145	Fl.Disc 250k	X	X	1	CU for 4 drives
P830-150	Fl.Disc 1M	X	X	1	CU for 2 drives
	V24	X	-	1	integrated on P854
P827-140	X1250 Disc	-	X	1	CU for 4 drive
P847-120	IPL CU	X	-	1	Remote loading CU
P845-145	ASCU4Z	X	X	1/2	

The connection of non-standard devices to the system must also be made via a control unit. Standard boards are available on which the customer may build his own control unit. Boards are available without any logic circuits (printed circuit boarding) or with standard address and function decode logic and interrupt encoding already mounted and connected (general purpose cards).

INPUT/OUTPUT TYPEWRITERS

P842-001 PER3100 Matrix Printer

Figure 15.1 shows the P842-001 matrix printer and keyboard with V24 interface.

The P842-001 matrix printer and keyboard offers the same basis facilities as the typewriter without attached paper tape equipment. It is capable of near silent operation at up to 50 characters per second and may use peg or friction fed paper of various widths, multiple copies being available when peg fed paper is used.

Figure 15.1

Line spacing of 1, 1 1/2, or 2 normal lines and LOCAL/ON/LINE/OFF operation are selectable at the printer. Various keyboard layouts and character sets are available, including the possibility of up to 7 special characters on option.

Connection to the System

Connection to the system may be via the programmed or input/output processor channel and is made via a V24 serial control unit.
In all cases the maximum printer speed is 50 characters per second although the actual speed of transfer will depend on the control unit, interfacing, and program being used. The available interface boards enable transfer speeds of 100-9600 baud to be selected in specific steps. Where transfer rates of above 50 characters per second occur or in the case of certain special characters the controlling program must insert sufficient null characters to avoid the loss of data.

Main Controls

Power On/Off Switch - An external two position switch, used by the operator to switch the mains power to the printer On or Off.

Operational Switch - An external two position switch, used by the operator in certain cases to make the printer operable.

Continuous Line Feed Switch - An external spring loaded switch, which whilst depressed causes line feeding of the paper to occur continuously.

Apart from the mentioned switches internal links exist on the standard interface boards within the printer, for the selection of line speed and to enable an echo print facility if this is required.

Basic Specifications

Operating Speed	- Up to 50 characters per second
Size	- Width 510 mm, height 170 mm, Depth 310 mm, without keyboard, 465 with keyboard.
Weight	- 20 Kilograms
Paper Width	
Peg Fed	- 231.8 mm, 203.2 mm and 314.3 mm (perforation distance).
Friction Fed	- 148 mm to 306.3 mm.
Power	- 100 VA Average.
Operating Temperature	- 10^{o} - 40^{o}C operating.
Relative Humidity	- 20 - 80%

P842-002

This is the same printer as the P842-001 but with current loop interface and the following additional switches:

Line Spacing Switch - A three position switch mounted on the KSR interface board and used to select the required line spacing when the KSR interface board is fitted.

Mode Switch - A three position switch mounted on the KSR interface board within the printer and easily accessible by the operator. The switch is used to control the mode of operation of the printer when the KSR interface board is used:

OFF Printer does not accept either line or keyboard inputs. The main power supply to the printer is not effected by this switch and may be ON.

LOCAL Printer interface is operable from the keyboard only, no line signals are sent or accepted by the printer.

ON LINE Printer interface is operable and may accept both line and keyboard inputs. Keyboard inputs are also retransmitted as line output.

P842-003 As P842-001, but without keyboard.

P842-004 As P842-002, but without keyboard.

PUNCHED TAPE EQUIPMENT

P801-001 Punched Tape Reader

Figure 15.2 shows the P801-001 Punched tape reader.

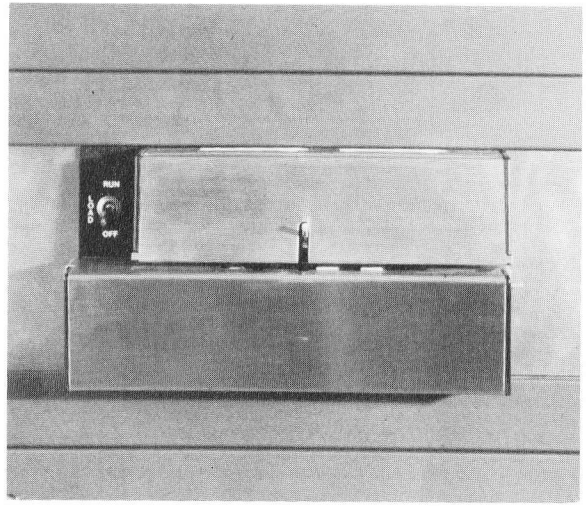


Figure 15.2

The P801-001 punched tape reader provides the system with the ability to read a wide range of punched paper tapes at a speed of up to 333 characters per second.

The reading assembly is of the photo-electric type and raises data and timing signals at TTL levels, 8 data channels and 1 timing channel being available. The tape drive unit controls the movement of the tape across the readhead via a drive motor and associated pinch roller and brake assemblies. No adjustment to the pinch roller is necessary when tapes between 0.064 to 0.124 mm (0.0025"to 0.005") thick are used and adjustment for 21.4 mm, or 25.4 mm (7/8" or 1") wide tape is carried out by an externally mounted control.

Connection to the System

Connection to the system may be via the programmed channel or an input/output processor.

Mounting

The complete reader, including power supply is assembled for mounting in a standard 19" rack and may be fitted into either the basic or an extension cabinet.

Main Controls

Power Switch - A three position switch mounted on the front panel of the reader, used for switching the power on the reader:

OFF — No power is switched on to the reader.

LOAD — Power is supplied to the drive unit motor and reading unit, the pinch roller and brake assemblies are clear of the tape track to allow loading.

RUN — Power is supplied to all the reader circuits and the reader operates under the control of the system.

Tape Width Selector - An adjustable control mounted on the side of the reader. The control is lockable and is used to adjust the tape guide mechanism as required.

Tape Load Lever - An external control on the front of the reader, used to disengage the front tape guide and allow insertion of the tape.

Basic Specifications

Operating Speed	- 333 characters per second
Size	- Width 483 mm, Height 133 mm, Depth 203 mm.
Weight	- 15 Kilograms
Tape Size	- Width 21.4 mm, 25.4 mm (7/8", 1") selectable. - Depth 0.064 to 0.124 mm (0.0025" to 0.005")
Power	- 150 VA
Operating Temperature	- 0 - 45^{o}C.
Relative Humidity	- 20 - 80%

P803-001 Paper Tape Punch

Figure 15.3 shows the P803-001 Paper Tape Punch.

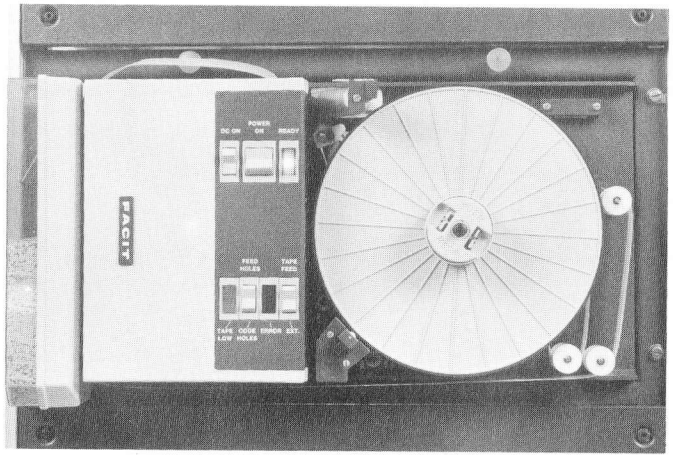


Figure 15.3

The P803-001 paper tape punch provides the system with the ability to produce a punched paper tape output at a rate of up to 75 characters per second on various width tapes. No adjustment is necessary for tapes of 0.08 to 0.11 mm (0.0031" to 0.047") thickness and the punch may be set to accept tape of between 17.5 mm 11/16") and 25.4 mm (1") in width. Both supply and take up bobbins are fitted and can be used with reels of tape up to 20 cm in diameter. The punch includes its own power supply.

Connection to the System

Connection to the system may be via the programmed channel or an input/output processor.

Mounting

The punch is available assembled for mounting in a standard 19" rack or as a free standing unit.

Main Controls and Indicators

Power On Switch - A two position switch mounted externally, used to switch the mains power to the punch On or Off.

DC On Switch - A two position switch mounted to one side of the Power On switch, used to switch the internal d.c. supply to the punch.

Tape Feed Switch - A two position switch mounted externally and spring loaded to the off position. When the switch is depressed tape is fed from the supply reel to the take up bobbin without punching.

Feed Holes/code Switch - A three position switch mounted externally and spring loaded to the central, off, position. When the switch is depressed tape is fed from the supply reel to the take up bobbin and either feed holes only or feed holes and code holes in all tracks are punched, with respect to the depressed position of the switch.

Apart from the main controls, indicator lights are mounted externally to indicate: d.c. power on, tape supply low, and certain errors. Internal switches are also fitted to control the take up bobbin.

Basic Specifications

Operating Speed	- 75 characters per second
Size	- Width 330 mm, height 190 mm, Depth 432 mm.
Weight	- 13 Kilograms.
Tape Size	- Width 17.5 to 25.4 mm (11/16 to 1") Thickness 0.08 to 0.1 mm (0.0031" to 0.0043")
Power	- 180 VA max.
Operative Temperature	- 0 - 45^{o}C operating.
Relative Humidity	- 20 - 80% operating.

CARD READER

P806-102 Card Reader

Figure 15.4 shows the P806-102 card reader.

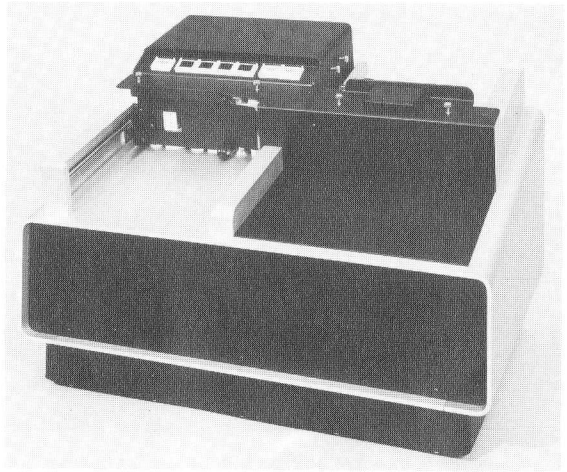


Figure 15.4

The P806-102 card reader provides the systen with the ability to read data from 80 columns cards at a transfer rate of up to 300 cards per minute. Card handling facilities in the form of an input hopper and output stacker enable the reader to handle up to 1000 cards without operator intervention for loading. The reader is of the photo electric type and employs a straight through card track with a vacuum picking mechanism, providing almost jam free operation and extremely long card life.

The reader is free standing and includes its own power supply.

Connection to the System

Connection to the system may be via the programmed channel or an input/output processor.

Main Controls

Power On/Off Switch - A two position switch mounted externally on the back of the reader, used to switch the mains power to the reader On of Off.

Mode Switch - A three position switch mounted externally on the back of the reader, used to select the mode of operation of the reader:

OFF	The reader is inoperative.
LOCAL	The reader is operative under the control of the operator.
REMOTE	The reader is operative under the control of the system.

Reset Switch - a push button switch mounted externally on the front of the reader, used to start or restart the reader in certain modes.

Stop Switch - A push button switch mounted externally on the front of the reader, used by the operator to stop the reader as required.

Apart from the main controls, lamps are provided to indicate the state of the reader and other switches are provided for the testing of the lamp and the setting of the reader for automatic or manual shutdown when necessary.

Basic Specifications

Operating Speed	- 300 cards per minute.
Size	- Width 58.6 cm, Height 41.2 cm, Depth 45.7 cm.
Weight	- 34.4 Kilograms.
Card Specifications	- Standard 80 column card.
Power	- 1650 VA starting, 600 VA running.
Operating Temperature	- 15 - 25^{o}C Limits imposed by cards.
Relative Humidity	- 50 - 70%.

LINE PRINTERS

P809-002/P809-004 Matrix Line Printer

Figure 15.5 shows the P809-002 matrix line printer.

Figure 15.5

The P809-002 matrix line printer provides the system with the ability to produce a printed output at a rate of up to 200, 132 column lines per minute on standard fan folded paper, with a character set of 72 characters. Where necessary an output can be to a preset format and adjustment is possible to accomodate a paper width between 100 and 440 mm.
The P809-004 has the same characteristics, except that it produces up to 400 lines per minute.

The carriage is a shuttling bar mounted on a support which moves in the horizontal plane between two side plates.
The printer is a free standing unit and includes its own power supply.

Connection to the system

Connection to the system may be via the programmed channel or input/output processor.

Main Controls

POWER-ON - A pushbutton indicator holding switch mounted externally used to switch the main power to the printer on and when pressed again, off.

PAPER FAULT (P809-004 only) - An indicator which is illuminated when a paper fault condition is detected.

START/STOP - A pushbutton momentary indicator switch mounted externally. When pressed the indicator is lit and the printer is operational. When pressed again the indicator light is extinguished and the operator can use the TOP OF FORM and SINGLE LINE Pushbuttons.

TOP OF FORM - A pushbutton momentary switch mounted externally whose action is inhibited when the START/STOP button is lit. When pressed in STOP mode the paper is advanced to the next top of form position.

SINGLE LINE - A pushbutton momentary switch mounted externally whose action is inhibited when the START/STOP button is lit. This pushbutton allows to advance the paper one line.

ERROR - An indicator which is lit when an error condition occurs.

Basic Specifications

Operating Speed	- P809-002 : 200 lines per minute - P809-004 :400 lines per minute
Line Length	- 132 characters.
Size	- Width 700 mm, Height 800 mm, Depth 460 mm.
Weight	- approx. 80 Kilogram.
Paper Specification	- Single copy 15 lb bond min. Multiple Copy up to 5 parts 11 lb bond with interleaved carbon. Paper width 100 - 440 mm.
Power Consumption	- 300 VA.
Operating temperature	- 10 - 40^{o}C.
Relative Humidity	- 20 - 80%.

MAGNETIC DISC EQUIPMENT

P824-012 Moving Head Disc Unit

Figure 15.6 shows the P824-012 moving head disc unit.

Figure 15.6

This moving head disc unit provides the system with the ability to transfer data to or from a magnetic disc cartridge at a rate of up to 312k characters per second after initial access to the disc area required. Head positioning is carried out by an electro-mechanical mechanism with a positive positioning detent, the average positioning time being only 30 ms. Data are recorded on or read from the disc serially, 6k characters (or bytes) per track being possible. Each surface of the disc contains 200 tracks providing an overall capacity of 2.7M characters. A second, fixed disc is incorporated within the unit. The moving heads for this are combined within the overall head mechanism and thus the capacity, transfer rate and access time are all the same as for the exchangeable cartridge, overall capacity of the unit with the fixed disc being 5.4M characters. Apart from the drive and head position mechanism the unit contains its own power supply and all the necessary control logic for correct operation.

The exchangeable recording disc is a Philips 14" mono disc cartridge, P824-100, and is fully compatible with the IBM 5440 type of cartridge with 16 sectors.

Connection to the System

Connection to the system is via an input/output processor.

Mounting

The complete unit is assembled for mounting within a standard 19" rack by means of mounting kit P849-039 and fitting may be in either the basic or an extension cabinet. The drive unit is slide mounted within the rack to enable cartridge changing and engineering maintenance.

Main controls

All the main controls are mounted externally on the front panel of the unit and incorporate their own indicators.

Power On/Off Indicator - Is lit when the power is switched on.

Start/Stop Switch - A push button switch used to start and stop the drive

Apart from the main controls indicator lamps are fitted on the front panel to indicate Cartridge Exchange and certain fault conditions.

Basic Specifications

Operating Speed	- Disc rotation, 2400 r.p.m.
	- Transfer, 312k characters per second.
Average Latency	- 12.5 msec.
Average Access time	- 33 $\pm$ 2 msec.
Size, Drive Unit	- Width 480 mm, Height 262 mm (3U)
	Depth, 752 mm.
Weight Drive Unit	- 60 Kilograms.
Power	- 150 VA running (600 VA starting).
Operative Temperature	- 10 - 35^{o}C.
Relative Humidity	- 20 - 80%

P824-014 Moving Head Disc Unit

The P824-014 Moving Head Disc Unit is functionally identical to the P824-012, but offers a capacity of 5.4M characters on the removable disc and 5.4M characters on the fixed disc.

P827-001 Fixed Disc Drive

The P827-001 fixed disc drive unit provides the system with a 6M bytes external storage device.
The disc drive uses a single 14" disc as storage medium, based on Winchester technology. The applied recording method is modified frequency modulation (MFM), with a data rate of 2M bits/sec. A formatted disc contains 52 sectors of 256 bytes per track, which gives an overall capacity on 2 * 225 tracks of 5.99 Mbytes.

Connection to the System

The P827-001 is connected to the system via an I/O processor. Up to 4 drives can be connected to one control unit, in daisy-chain configuration.

Mounting

The P827-001 is mounted in an equipment shlef, which fits into the standard 19" rack. The power supply for the drive is contained in the equipment shelf.

Main Controls

The P827-001 disc drive has no controls or switches accessible by the operator.

Basic Specifications

Number of discs	- 1, diameter 14"
Recording surfaces	- 2
Number of heads	- 2
Number of physical tracks per surface	- 231
Number of spare tracks per surface	- 6
Number of useful tracks per surface	- 225
Recording Method	- MFM (modified frequency modulation)
Disc speed	- 720 r.p.m.
Average latency	- 42 msec
Seek time (without settling time)	
track to track	- 2.5 msec
avergae	- 192.5 msec
maximal	- 575 msec
Settling time	- 40 msec
Track density	- 100 tracks per inch
Data rate	- 2M bits/sec

FLEXIBLE DISC EQUIPMENT

Figure 15.7 shows two flexible disc drives in an equipment shelf.

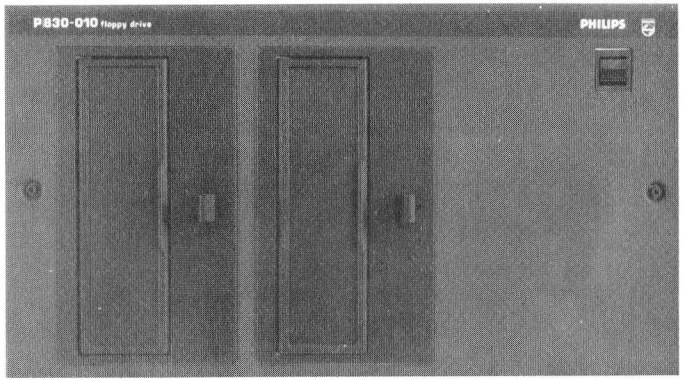


Figure 15.7 Flexible disc drives

Four types of flexible disc drives are available:
P830-006 flexible disc drive, max. 250k bytes, without doorlock
P830-015 flexible disc drive, max. 250k bytes, with doorlock
P830-025 flexible disc drive, max. 1M bytes, without doorlock
P830-035 flexible disc drive, max. 1M bytes, with doorlock

Connection to the System

The flexible disc control unit may be connected to the system via the programmed channel or an input/output processor. Two control units are available: control unit P830-045, which can control up to four P830-006 or P830-015 drives, and control unit P830-150, which can control up to four P830-006 or P830-015, or up to two P830-025 or P830-035.

Mounting

Up to two disc drives may be mounted in one 19" equipment shelf with self contained power supply for two drives.

Main Controls/Indicators

- Power switch: a two-position tumble switch with indicator on the equipment shelf's front panel. When lit, the power is switched on.
- Disc access door knob: when the door is closed, pushing the knob to the right will open the spring-loaded door.
- Door lock indicator (on P830-015 and P830-035 only): an indicator next to the door; lit when the door is locked.

Flexible Disc Hardware Format

A 1MB flexible disc may have any one of three hardware formats, known as Format 0, Format 1 or Format 2; a 250kB flexible disc may only have Format 0. The first cylinder of every disc is in Format 0, as it contains information about the size and format of the rest of the disc. All formats are IBM compatible. The following table summarises the formats:

	Format 0	Format 1	Format 2
number of surfaces	1	2	2
number of tracks/surface	74	74	74
sectors/track	26	26	8
bytes/sector	128	256	1024
capacity (bytes)	250k	1M	1.2M
density (single/double)	S	D	D
max. transfer	1664	3328	4096

Basic Specifications

	P830-006/015	P830-025/035
Max. capacity	- 250k bytes	- 1.2M bytes
Number of heads	- 1	- 2
Number of tracks (max.)	- 77	- 77
Transfer rate	- 250k bps	- 500k bps
Rotational speed	- 360 r.p.m.	- 360 r.p.m.
Av. latency time	- 83 msec	- 83 msec
Av. seek time	- 260 msec	- 96 msec

DISPLAY EQUIPMENT

P817-001/002 Display Units

Figure 15.8 shows the P817-002 display.

Figure 15.8

The P817 provides the system with a table-top terminal which displays its information on a 12" screen, 80 characters per line. The number of lines displayed on the screen is 24.

The P817-001 is character-oriented and has a basic character set of 64 characters, optionally extendable to 96. The character set extension consists of lower case facility.
The P817-002 is character- or block-oriented and has a basic character set of 96 characters.

The keyboard is detachable. For P817-001 a choice can be made between two keyboards, with or without numeric pad and extended character set. For P817-002 one keyboard is available with the 96-character set, numeric pad and additional control functions.

Connection to the System

Connection to the system is done via standard serial V24 interface, an optional plug-in current loop interface or an optional plug-in in-plant modem.

Controls

POWER ON/OFF: a combined pushbutton and indicator.
ON LINE/OFF LINE: a switch to connect the display to the system and disconnect it.
BRIGHTNESS CONTROL: a thumbwheel to adjust the brightness of the characters on the screen

An attractive feature of this display is that a number of functions are soft-selectable.
That is to say, for these functions a certain value is preset by means of straps, but the user has the possibility of selecting alternative values by means of a simple keyboard action, or, for P817-002, also by software.
These functions are:

- Baud rate
- Half duplex/full duplex
- parity odd, ever or none
- 1 or 2 stop bits
- Size split screen
- Scroll yes or no
- Character or block mode

Basic Specifications

Number of lines	- 24
Number of Chr./line	- 80
Transmission rate	- up to 9600 bits/sec. (selectable)
Transmission mode	- asynchronous
Parity selection	- odd, even or none
Interface	- V24, current loop or inplant modem
Refresh rate	- 50 Hz
Size	- display: 340 x 270 x 380 mm display on swivel stand: 340 x 430 x 380 mm keyboard: 500 x 85 x 180 mm
Weight	- display: 17 kg. swivel stand: 8 kg. keyboard: 5 kg.
Power	- 175 VA
Operating Temperature	- 5 - 40^{o}C
Relative Humidity	- 5 - 95% (non-condensing)

16 Software

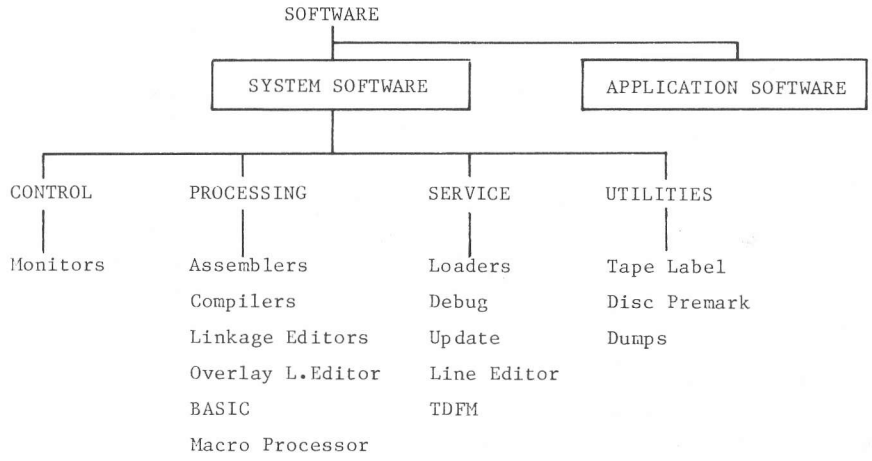


Figure 16.1 Standard System Software and Application Software

Software consists of two main divisions, as shown by figure 16.1. The application software shown is not further divided as this represents the programs a user writes to carry out his processing requirements, these of course will vary considerably from user to user. System software consists of all those programs a user may employ to efficiently produce and execute his application software and is made up of the control, processing, service and utility programs shown by figure 16.1. Full descriptions of the available system software are covered in the appropriate software manuals and therefore only a brief description of the software is given here.

All software may be of a modular construction, and in the case of the monitors, a user may select the modules he requires at the time of generating the system. The main advantages offered by modular programming are:

1. Modules may be written in different source languages and by different programmers thus enabling and efficient and speedy solution to any problem.
2. Common routines may be written and held for use by a number of programs.
3. Testing, error detection and correction procedures are simplified.
4. Updating is simplified.

Figure 16.2 shows the method of modular construction used within the monitors as an example of modular programming. Each monitor may be assembled from only those modules it requires.

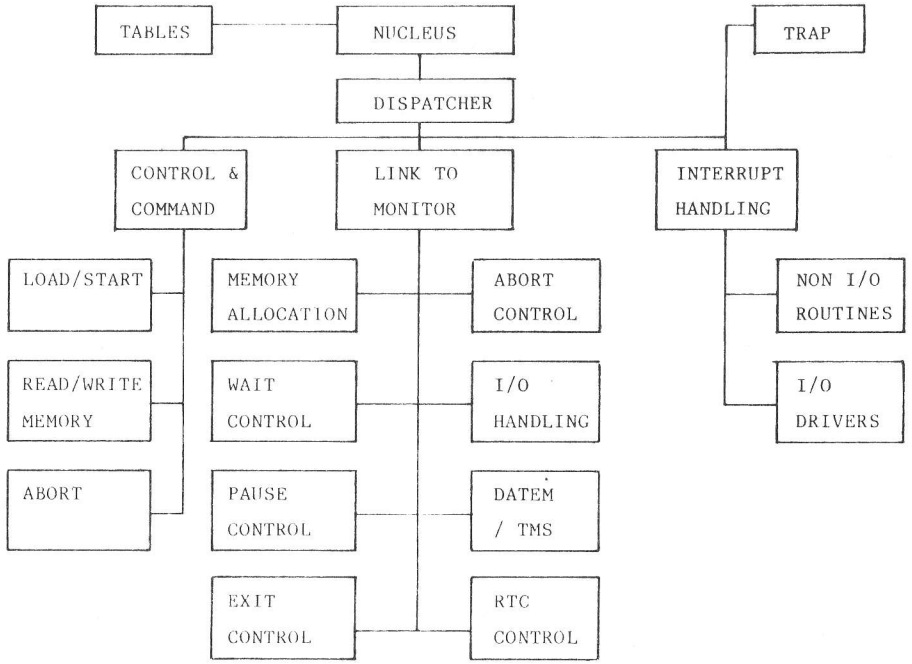


Figure 16.2 Modular structure of monitor

Figures 16.3 to 16.6 show the standard system software configurations.

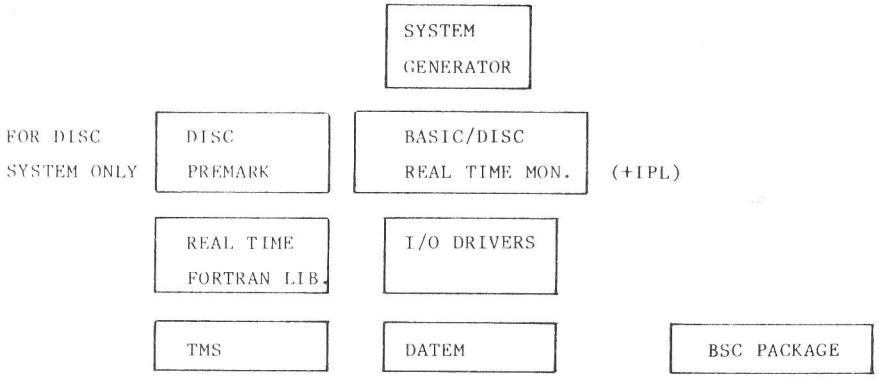


Figure 16.3 Software for Basic and Disc Real Time System

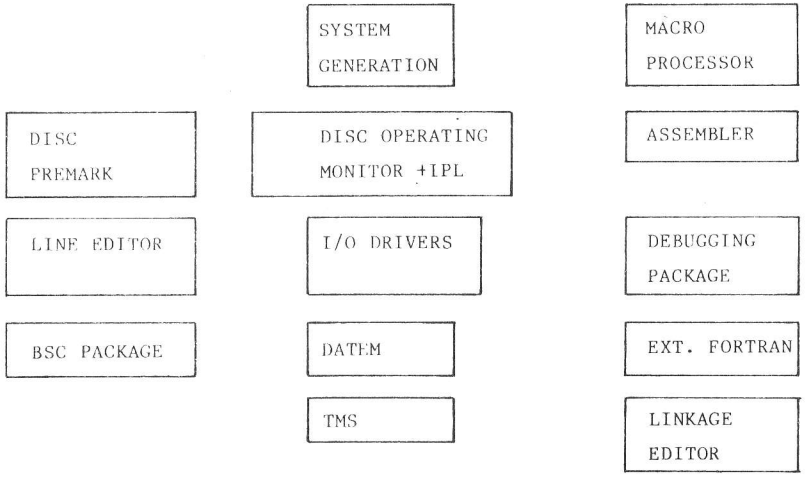


Figure 16.4 Software for Disc Operating System

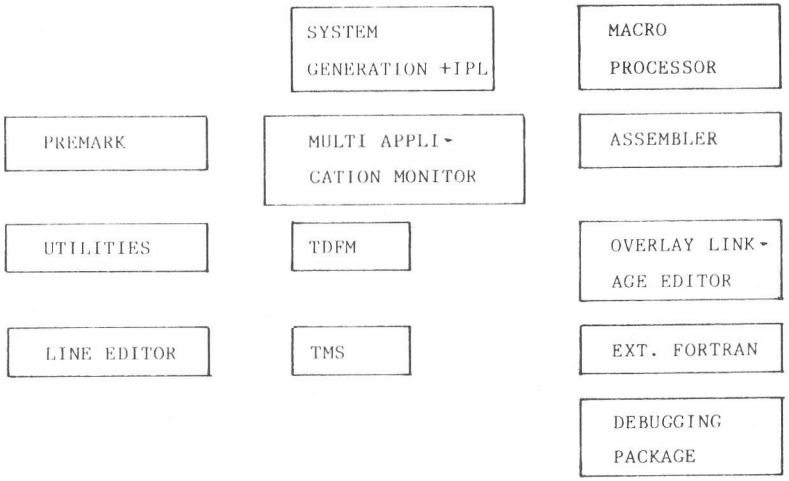


Figure 16.5 Software for Multi Application System

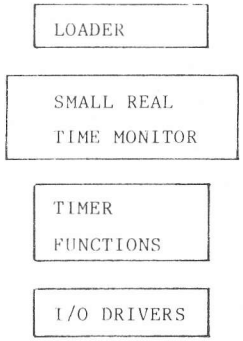


Figure 16.6 Software for Small Real Time System

CONTROL PROGRAMS

The loading, preparation and supervised execution of application and certain system software may be handled by control programs, known within system software as monitors. The user may select only those modules which he needs when generating his system, and if necessary expand the monitor facilities as and when the hardware system is enhanced.

There are five monitors currently available, one for each of the software configurations that require a control program. The monitors available are:

1. Basic Real Time Monitor.
2. Disc Operating Monitor.
3. Disc Real Time Monitor.
4. Small Real Time Monitor.
5. Multi Application Monitor.

Basic Real Time Monitor

The Basic Real Time Monitor (BRTM) is intended to supervise the execution of user application programs in a real-time environment. It provides no development facilities, assuming that the user will develop and pretest his routines under another operating system.

The system is based on a priority structure, comprising a maximum of 48 hardware interrupt levels and up to 15 software user levels. Routines running at a low level number have a higher priority than those running at a higher level number.

When loaded into a machine fitted with a Memory Management Unit (MMU), the Basic Real Time Monitor can handle the running of programs in up to 128k words of memory. When no MMU is available the maximum memory size which can be handled is 32k words.

The monitor has a modular structure, so that the user may easily modify it to his own requirements. At run time there is little noticable difference between the user's and the system's routines; either may run at any level. Several routines may run at the same software level, giving a measure of multi-programming at that level; the use of "scheduled label" routines further increases the possibilities.

The minimum configuration required for BRTM is a CPU with 8k words of memory, an operator's typewriter, and a paper tape reader.

Disc Operating Monitor

The Disc Operating Monitor (DOM) is intended mainly as a tool for program development in a batch or interactive environment; it is not a real-time operating system.
A comprehensive set of control commands allow communication with the system, normally through the operator's typewriter, and all the peripherals may be used without restriction, including the discs. The monitor controls the runnning of only one program at a time.

The monitor allows the user to create, process and maintain on disc all kinds of user data, such as programs in source, object and loadable forms as well as various data files. This can be done in batch or interactive mode.

The scheduled label feature allows some form of multi-programming, and the catalogued procedure feature permits the user to store commonly used sequences of control commands and execute them when required.

The minimum configuration required for DOM is a CPU with 32k words of memory, an operator's typewriter, and one disc unit.

The Disc Operating Monitor can be used in systems equipped with felxible discs; in that case the monitor is stored on a flexible disc, and the I/O driver contained in the monitor simulates an X1215 disc (i.e. standard disc file management can be used for files stored on flexible discs).

The minimum configuration required for DOM on flexible disc is a CPU with 32k words of memory, an operator's typewriter, and two flexible disc drives.

Disc Real Time Monitor

The Disc Real Time Monitor (DRTM) is a disc-based system, intended to supervise the execution of user application programs in a real-time environment. It provides no development facilities, assuming that the user will develop and pretest his routines under the Disc Operating Monitor. Debugging facilties, however, are available.

The system is based on a priority structure, comprising a maximun of 48 hardware interrupt levels and up to 15 software user levels. Routines running at a low level number have a higher priority than those running at a higher level number.

The user may decide that certain routines need to respond very rapidly to some condition, or are used very often; these routines will be kept in memory. Other routines may respond more slowly, or are required only infrequently; these routines will be stored on disc and only loaded into memory when their execution is required.

The monitor has a modular structure, so that the user may easily modify it to his own requirements. At run time there is little noticable difference between the user's and the system's routines; either may run at any level. Several routines may run at the same software level, giving a measure of multi-programming at that level; the use of "scheduled label" routines further increases the possibilities.

When running on machines fitted with an MMU, DRTM can handle a number of user application areas, located in absolute memory addresses over 32k words, up to a maximum of 128k words.

The minimum configuration required for drtm is a CPU with 16k words of memory, an operator's typewriter, and one disc unit.

The Disc Real Time Monitor can be used in systems equipped with felxible discs; in that case the monitor is stored on a flexible disc, and the I/O driver contained in the monitor simulates an X1215 disc (i.e. standard disc file management can be used for files stored on flexible discs).

The minimum configuration required for DRTM on flexible disc is a CPU with 32k words of memory, an operator's typewriter, and two flexible disc drives.

Small Real Time Monitor

The Small Real Time Monitor is developed for dedicated computer applications requiring a small and fast monitor. The monitor is paper tape oriented.

Included in the monitor are timer functions for control of the user programs. Fourteen software priority levels allow multiprogramming between user tasks, one or more programs may be connected to the same level.
The monitor is upward compatible with the Basic Real Time Monitor.

Multi Application Monitor

The Multi Application Monitor is particularly well suited for a number of applications:

- multi tasking applications where a large memory size allows more resident programs or several transient programs to be in core, improving response time and overall performances.
- foreground/background applications where program debugging can be made concurrently with a real-time process.
- data communication applications where many buffers and tables have to be resident due to fast access time.

The Multi Application Monitor is a disc oriented monitor and has a clear open ended structure; at each application corresponds a machine defined by:

- several priority levels,
- several allocated or shared peripheral devices (spooling),
- several memory partitions,
- several disc file libraries.

More than one multi-tasking or real-time machine exists; programs can be connected to a real-time clock or timer, several disc resident programs can be in memory at the same time (multi transient areas).
A batch processing machine supports other system components including Assembler, Overlay Linkage Editor, Extended Fortran Compiler, Line Editor etc. This machine is specially oriented towards program development. Each machineand the monitor itself are individually protected. The Monitor can be extended with the Transaction orientended Disc File Management Package which adds to the existing system data base facilities with direct and sequential access, variable length data records, indexed organisation, file protection and on-line updating.

A set of operator commands is available e.g. to create or suppress a machine, to allocate or deallocate memory for a machine, to assign peripheral devices or files, to start or stop tasks inside machines.

The minnum configuration required for MAM is a CPU with 64k words of memory and an MMU, an operator's typewriter, one disc unit, and a line printer.

DATEM

DATEM is a data communication monitor extension on the Basic Operating Monitors, the Disc Operating Monitor and the Basic and Disc Real Time Monitors. It provides the system with basic data communication facilities. The standard features of the monitors remain available for the system. The extension takes care of the following functions in a data communication configuration:

- Connection to the line (leased lines and switched lines).
- Read or write data.
- Error control.
- Time-out control.
- Data control (wait for data, polling and selecting, stop the transmission on detection of special characters).

BSC

BSC is the Binary Synchronous Communication line procedure package which may be used for synchronous data communication. It handles the line control of the transmitting and receiving stations.

TRANSMISSION MANAGEMENT SYSTEM

The Transmission Management System (TMS) is a data communication subsystem specifically desiged for use in computer network environments. TMS is intended for use with the existing P800 operating systems, and enables application programs to transfer and receive data via DC lines.

TMS software has a layered structure which facilitates inter-processor communication, and has been designed to offer the following three-layer approach to network software:

- application layer - DC related application processing
- transport layer - network control
- communication layer - data exchange control

Within this layer structure, the TMS communication layer caters for four principal levels with the functions indicated:

- the physical level - control unit, modem
- the driver level - control unit DC control functions
- the procedure level - transport layer interface

At the TMS procedure level, both the High-level Data Link Control (HDLC) and the BSC protocols are supported. TMS modularity enables the introduction of new protocols without the necessity for interface changes. The HDLC procedure implemented within TMS is the X25 HDLC procedure (balanced mode).

PROCESSING PROGRAMS

These programs consist of assemblers, compilers, linkage editor, overlay linkage editor and macro processor, available to a user for the production of his application software. Various versions of the programs are possible to meet the requirements of different systems.

Assemblers

The assemblers convert source modules written in assembly language into object modules suitable for linking to other object modules or for loading and execution. Each line of a source module is written in assembly language and represents one central processor instruction, word or block of data or directive, to control the assembly process. Additional features available include error reporting and recovery, assembly listing and the selection of the peripheral devices to be used during processing.

Extended FORTRAN Compiler

The Extended FORTRAN compiler translates FORTRAN source programs into object modules to be processed by the Linkage Editor or Overlay Editor with the Mathematical Library. The result of the editing process is a self-contained executable program which can run under control of the monitor. The compiler is self-initializing and does not require reloading between successive compilations.

The compiler can produce modules for use with the Floating Point Processor or modules for simulated floating point operation, both in reentrant or non-reentrant code.

The source language for the Extended FORTRAN compiler complies mainly with the ANSI X3.9-1978 standard.

The Real Time FORTRAN system is a system in which user written FORTRAN programs run under control of a real time monitor. The Real Time FORTRAN library consists of a set of routines which are called by the FORTRAN program whenever their use is required.

Activity management System

The Activity Management System (AMS) runs under MAM and provides an environment for the user to write and run his own application programs.
The environment supports the interactive use of terminals to converse with the applicationa, the use of TDFM files, and communication between virtual machines; this may require the use of data communications in a network of two or more physical machines. This last component (AMSNET) may also be used by non-AMS users, if they do not interfere with the AMs users.

The user writes his application in Activity Management Language (AML), a proper subset of FORTRAN with extensions to handle the features mentioned above. The system includes an AML Processor to convert source AML to loadable programs, and generation utilities to tailor the "standard" software components to the user's own requirements.

RTL/2

RTL/2 is a high-level programming language for real time applications. It is especially suited for the programming of on-line data collection, communication and control systems.
The RTL/2 system consists of a compiler, which translates RTL/2 programs into assembly language source modules, a linkage verifier, which checks external references between separately compiled modules, and run-time support routines (e.g. for standard stream I/O).
An RTL/2 - FORTRAN interface allows FORTRAN subprograms to be called from RTL/2 modules.

Linkage Editor

The Linkage Editor is available for the Disc Operating Systems providing the facility to link separate object modules either for direct loading and execution, to be loaded later or used within a further linkage process. By linking, all the advantages of modular programming are easily available.

Modules which are to be linked are written containing specified external references and entry points to be used during linkage, and the control of the linkage process by the operator, allows for the selection of the peripherals and mode to be used during processing.
In addition the program provides the listing of a map reports errors during processing.

Overlay Linkage Editor

This processor runs under control of the Multi Application Monitor. It applies for large programs which cannot fit in the available partition. The Overlay Linkage Editor produces from a set of object modules a segmented program organised in an overlay structure which is transparent to the user. It satisfies all the external references and produces calling sequences for loading the different segments. In case of one segment the processor can also be used as a simple Linkage Editor.

After processing a load module is produced recorded on a temporary load file. This load module may be executed and/or kept in library.
A number of options may be typed in e.g. to specify an absolute loading address, the start address of a common area or the specification which library will be scanned, and some information required by the Debugging Processor.

Macro Processor

The Macro Processor generates text (e.g. assembly language source code, or data for a data file) from macro definitions and macro calls. The macro definition contains the text framework, the macro call specifies the macro definition to be used and provides parameters.
The Macro Processor runs under control of the Disc Operating Monitor or the Multi Application Monitor.

SERVICE AND UTILITY PROGRAMS

The programs within these groups provide the user with all the facilities required to set up, run, and maintain the system apart from the tasks of initial program production.

Debugging

Debugging programs are available to enable rapid error detection within program modules, and to provide the programmer with the ability to stop a program at specific points so that the contents of memory and/or registers may be checked or altered as necessary.

Line Editor

The line editor, available for use within the disc operating system configuration, provides all the facilities of the Update Package and an additional facility to enable the alteration of a specified character string wherever such a string appears in a module.

Utility Programs

These programs are used by the user during the setting up of the systems files, and where necessary during the normal running of the system to provide the marking and labelling facilities required by certain peripherals and the information required by certain processes.

BASIC

BASIC is a system for compiling and executing programs written in the BASIC (Beginners All-purpose Symbolic Instruction Code) language.
The BASIC system comprises four system components:

- the kernel component, for single user program development and execution
- the real time component, for real time applications
- the business component, for decimal data processing and disc file handling
- the multi-user component, for up to 16 users working simultaneously

BASIC programs are translated into an intermediate code by a compiler; at run time the intermediate code is interpeted and executed statement by statement by an interpreter.

The following combinations of BASIC system components and P800 monitors used:

- DOM & kernel component
- BRTM & kernel component
- DRTM & kernel component
- BRTM & real time component (including kernel component)
- DRTM & real time component (including kernel component)
- DRTM & multi user component (including kernel component)
- DRTM & business component (including kernel component)
- MAM & multi user component (including kernel component)
- MAM & multi user component (including kernel component) & business component

Appendix A IPL Parameters

The ROM bootstrap contained in the CPU loads the IPL via the device and channel, specified by means of parameters entered via the control panel. The following parameter bit values apply:

bit	value	meaning
0	0	character transfer via Programmed Channel
	1	word transfer via Programmed Channel
1	0	sequential device, ROM or 40M/80M disc
	1	disc device other than 40M/80M disc
2	0	no seek operation (fixed head disc or floppy disc)
	1	seek operation (moving head disc)
3	0	device connected to I/O Processor
	1	device connected to Programmed Channel
4-7		additional information for control unit:
		0011 X1215, X1216 or X1250 disc
		0000 floppy disc
		0000 punched tape
		0101 remote loading
8	0	single device control unit
	1	multiple device control unit
9	0	all devices except for X1215/X1216 disc
	1	X1215/X1216
10-15		device address

Hexadecimal parameter values for standard configurations are given on the next page.

The following parameter values apply to standard configurations:

1020	punched tape
103A	remote loading
3000	ROM
4083	floppy disc 250k on I/O Processor
6083	floppy disc 1M
63C1	X1250 fixed disc
63C2	X1215, removable disc
63E2	X1215, fixed disc
D083	floppy disc 250k on programmed channel

Comment Sheet

P854M System Handbook (5122 991 31231)

Name: _____

Company: _____

Department: _____

Address: _____

_____

Telephone Number: _____ ext. _____

Comments or Suggestions :

PHILIPS DATA SYSTEMS

MARKETING GROUP SMALL COMPUTERS
P.O. Box 245, Apeldoorn, The Netherlands
Phone: 055-330123; telex: 49142

For further details contact the above address or:

Sweden
Svenska AB Philips
Data Systems
Minidatorer
Rissneleden 16
Stockholm 27
Tel. 08 830300

France
Philips Data Systems
Département Mini-ordinateurs
5 Square Max-Hymans
75015 Paris 15
Tel. 01 734 7759

Switzerland
Philips AG
Data Systems
Binzstrasse 18
8027 Zürich
Tel. 01 442211

Norway
Norsk A/S Philips
Data Systems Division
Nils Hansens vei 2
Oslo 6
Tel. 02 679380

Western Germany
Philips Data Systems GmbH
Weidenauerstrasse 211/213
Siegen-Weidenau 21
Tel. 027 14041

Great Britain
Philips Data Systems
Elektra House
2 Bergholt Road
Colchester C04-5AA
Essex
Tel. 206 5115

Belgium
Philips Data Systems SA
Marketing Group Small Computers
Anspachlaan 1
1000 Brussel
Tel. 02 2193900

Austria
Philips Data Systems GmbH
Untere Donaustrasse 11
A1020 - Wien 2
Tel. 0222 2475610

The Netherlands
Philips Data Systems Nederland B.V.
Bordewijkstraat 4
Rijswijk (Z-H)
Tel. 070 906720